A Hybrid Technique for Network Structured Multiple Deformable Objects Tracking

Vijayashankar Sethuraman

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

December 2010

Abstract

In this thesis, a novel hybrid approach for tracking variable numbers of network-structured deformable objects is presented. The hybrid technique developed is a combination of the Network Snakes parametric active contours technique, and the Reversible Jump Markov Chain Monte Carlo (RJMCMC)-based particle filter approach. Additionally, a novel method for (semi-)automatic initialization of the network snakes is implemented.

The proposed technique is applied to the real biological problem of tissue-level segmentation, and automatic tracking, of a network of cells in confocal images showing the roots of the model plant Arabidopsis thaliana. The Network Snakes component is used to model the structure of cells in Arabidopsis roots, which are clustered together and delineated by shared object boundaries forming a network topology. The RJMCMC tracker is allowed to track the network node points over image sequences, and these tracked nodes are then used to control and re-parameterise the topology of the network snakes at each time step. This is followed by energy minimization of the network snakes which refines the tracked nodes and cell boundaries to settle at the energy minimum. Thus the component techniques complement each other in the hybrid approach.

A novel method of evaluating such network-structured multi-target tracking is also presented in this thesis, and is used to evaluate the developed tracking framework for accuracy and robustness using several real and synthetic time-varying and depth-varying (z-stack) image sequences of growing Arabidopsis roots. These evaluation results are presented, along with a comparison between the results of the independent component techniques and the hybrid approach. The results show that the hybrid

approach performed consistently well at all levels of complexity and better than the component methods alone.

With the spread of systems approaches to biological research, demand for such methods and tools capable of extracting quantitative measurements of biological samples from individual and time-based sequences of microscope images is increasing. To this end, a software tool has been developed that implements the proposed hybrid technique which is being used at the Centre for Plant Integrative Biology to extract model parameters such as cell counts, cell boundaries and cell wall elongation rates.

Publications

1. V. Sethuraman, A. French, D. Wells, K. Kenobi, and T. Pridmore, Tissue-Level Segmentation and Tracking of Cells in Growing Roots, *In Press, Machine Vision and Applications*, 2011

2. V. Sethuraman, S. Taylor, A. French, D. Wells, and T. Pridmore, "Segmentation and Tracking of Confocal Images of Arabidopsis thaliana Root Cells using Automatically- Initialized Network Snakes", *3rd International Conference on Bioinformatics and Biomedical Engineering*, 2009, pp. 1-4

3. V. Sethuraman, A. French, D. Wells, and T. Pridmore, Segmentation and Tracking of Arabidopsis thaliana Root Cells using Automatically-Initialized Network Snakes, *Poster presentation*, *11th International Conference on Systems Biology*, 2010

4. V Sethuraman, AP French, DM Wells, J Twycross, TP Pridmore, Segmentation and tracking of Arabidopsis thaliana root cells using automatically initialized network snakes. *Poster presentation*, *4th BBSRC Systems Biology Grant Holders' Workshop. Edinburgh*, 2010

5. V. Sethuraman, A. French, D. Wells, and T. Pridmore, Segmentation and tracking of networks of Arabidopsis thaliana cells through confocal laser microscope images, *Technical meeting At: British Computer Society, London*, *UK. - BMVA Microscopy Image Analysis for Biomedical Applications symposium*, 2010

Acknowledgements

I would like to thank my supervisors, Dr. Tony Pridmore and Dr. Andrew French, for their excellent guidance and invaluable advice throughout my PhD. Without their tireless support and ideas, this thesis would not have seen the light of day.

I would also like to thank Dr. Darren Wells at the Centre for Plant Integrative Biology, for his invaluable help in acquiring the images used in this thesis.

I would like to thank everyone in the University of Nottingham and the Centre for Plant Integrative Biology, who has helped me in various ways during the course of my PhD. A special thanks to Prof. Ian Dryden for devoting his valuable time and advice during the discussions on Reversible Jump Markov Chain Monte Carlo algorithm.

I would like to thank the British Council, especially the UK-India Education and Research Initiative (UKIERI) board, for their financial and moral support throughout my PhD.

Personally, I wish to thank my family for always being there for me and supporting me. This thesis is dedicated to them.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

The aim of the work reported in this thesis is to develop a novel technique for tracking network-structured multiple deformable objects. Tracking multiple objects is a challenging task, especially if the target objects are clustered and connected together. The connected targets share image features between them so that any motion or deformation of one object is dependent on, and can have an impact on its neighbours. The close proximity, interdependency and the topological arrangement of the target objects require a high-level global representation that explicitly captures the connections between the target objects. A tracking framework for such complicated structures should ideally be able to cope well with the network deformations and translations which are common to the domain. This thesis explores novel ways of combining existing techniques to develop a novel hybrid approach for segmenting and tracking network-structured multiple deformable objects. To demonstrate the ability of the proposed technique, it is applied to the real biological problem of tissue level segmentation and automatic tracking of networks of cells in confocal images of live roots of the model plant *Arabidopsis thaliana*.

Another contribution of this thesis is a novel method for evaluating network-structured multi-target tracking, which is then used to evaluate the developed

tracking framework for accuracy and robustness using real and synthetic time-lapse sequences of growing Arabidopsis roots.

1.1. Motivation for research

As the population continues to increase, the large-scale and largely unpredictable variations in growing conditions brought about by climate change, and the depletion of natural fertilizers such as phosphorus [1] [2] threaten to disrupt global food production. If an adequate food supply is to be sustained, it is vital that crops be developed which can react to, and flourish in, new, and probably harsher, environments. To achieve this, understanding plant growth, and the processes that affect plant growth, is of great importance. Hence, accurate quantitative data describing the growth of plants, their organs and cells is crucial. Quantitative growth data provides the link between plant properties and behaviours and the models of growth mechanisms needed to secure future crop production.

Traditionally, observing plant growth responses involved manually recording change, for example by physically marking specimens' size, etc. on their growth plates at set timepoints. Other approaches have marked the plants themselves to observe expansion effects; for example, marking roots with ink at fixed intervals [3] or embedding stained glass beads on the root surface [4] and observing the displacement velocities of these features.

In recent years, the adoption of digital image analysis-based approaches has allowed measurements to be made at the organ or whole-plant scale with much less manual interaction and much less obtrusively. Methods have been developed to assess properties of plant roots [5] [6] [7] [8], shoots [9] [10] [11], leaves [12] [13] [14] and seeds [15]. Typically, these tools require either digital photographs or scanned images of plant organs. The challenge is to create accurate, high-throughput systems which can gather data from large numbers of plants at frequent intervals while

keeping human interaction to a minimum and placing only minimal constraints on growing conditions [4].

The rapid development of confocal laser scanning microscopy (CLSM) has allowed biologists to examine plant growth at the cellular scale. A wide variety of fluorescent markers are available which can be used to tag features of interest, e.g. cell walls, nuclei etc., which can then be imaged to allow visualisation of the structures within a growing plant. Confocal microscopes gather data from a plane through their specimen; the resulting images can be arranged either as 3D image stacks or single or time varying sequences of 2D sections through the plant. They can then be used to resolve the geometry and/or growth of the underlying cells which make up the tissue, allowing quantification of parameters such as cell volume estimates, growth rates, and so on. Extracting such information is, however, a very labour intensive and error-prone process when done manually, even in two dimensions. As a result, there has been much recent interest in developing automated methods of extracting such information from these images [16] [17] [18]. Automated approaches both help to remove the subjectivity of manual methods, and save experimenters' time, as much less manual interaction is required.

Existing methods and tools which aim to automatically measure plant growth at a microscopic scale include RootFlow [19], which uses optical flow techniques to measure growth rates across a magnified root image. Using such methods, the growth rates are associated with texture flow across the image rather than being tied to particular cells or other semantic regions. This local motion approach has been extended for use on CLSM images [20] [21], where noise effects and the properties of the imaging modality themselves, such as the lack of stable texture, need careful consideration to extract reliable motion information.

While measurements of growth at the plant organ and individual cell levels are invaluable, many key events in the plant life-cycle are marked by changes at the tissue level; they involve not just individual, but sets of related cells. The emergence of new lateral roots is a prime example. Here new primordia force their way from deep inside the main root, breaking through overlying tissues to form the lateral roots [11]. Even simple growth is best understood by considering multiple cells. Cells are connected, and do not 'slide' past each other as the plant grows; a developing plant can and should be thought of as a network of cells, shifting and expanding over time, considering not just local but global information about growth and change.

The goal of the work reported here is to identify and track the movement and deformation of networks of cells of the root of the model plant *Arabidopsis thaliana*. Arabidopsis attracts much attention within plant science because of its fully sequenced genome and simple cellular structure; as a member of the Brassicaceae family, biological results obtained using Arabidopsis are expected to transfer to crop plants. There is much interest in measuring differences in growth rates, cell volumes, and cell geometries across various mutants and wild-type plants. Knowing volume changes over time yields information about growth processes on a cellular scale, and recovering the cellular and tissue geometry allows models to be reconstructed from the data.

1.2. Challenges involved

Figure 1 shows a typical confocal image of an Arabidopsis root, with key features marked. New cells are formed, by cell division, in cells surrounding the quiescent centre and, as the plant grows downward, these cells "move" into the elongation zone. Here, during normal growth, cells expand in a direction parallel to root orientation. Arabidopsis plants grow quickly in the elongation zone and so cells can appear to expand and translate at high speed, with large displacements between

images. A wide range of cell deformations is also possible, depending on plant phenotype and growing conditions. Again, these deformations may be large, and are largely unpredictable. While the high resolution of CLSM images gives an impression of quality to the naked eye, the noise associated with confocal images is more complex than usually encountered. In [20], it has been shown to comprise two additive Gaussian components and an element of salt-and-pepper noise. Attenuation of the laser by plant tissue further means that image quality deteriorates as depth through the sample increases. Extracting accurate descriptions of the development of Arabidopsis cells from sequences of CLSM images is a challenging task.

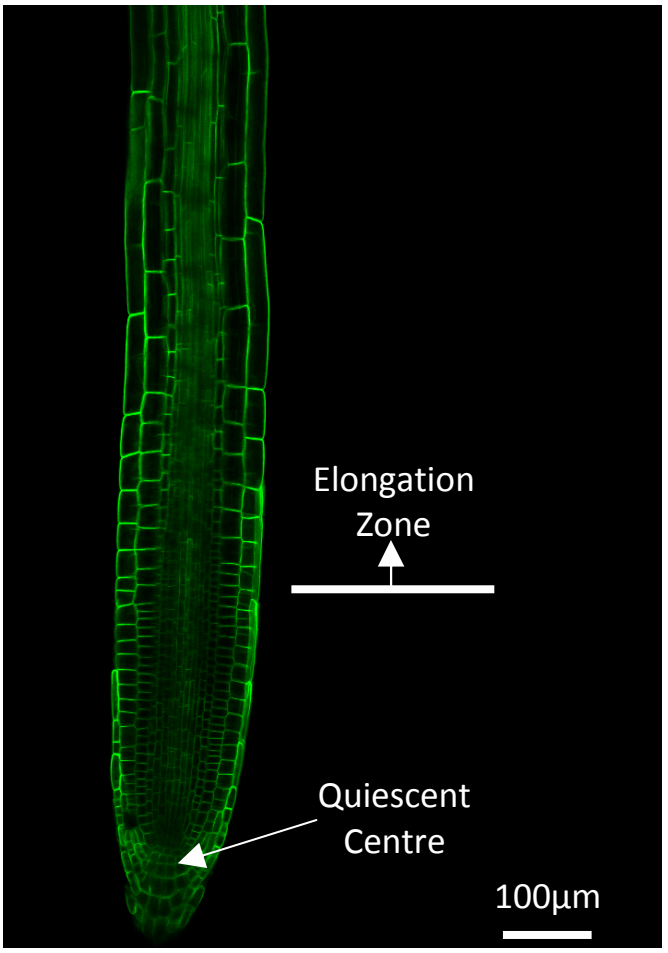


Figure 1. A typical CLSM image of the root of Arabidopsis, with the cell plasma membrane marked with green fluorescent protein and key features annotated.

As can be seen in Fig. 1, plant root cells are naturally clustered together, connected and delineated by shared object boundaries, forming a network topology. Explicit representation of this network has a number of advantages: it inherently encapsulates the underlying structure of the imaged tissue, it can be used to aid tracking of related cells by using neighbouring cells to help constrain estimates of cell position and geometry, and the resultant data structure is well suited for use in plant growth modelling frameworks.

The approach adopted here is based upon the 'Network Snakes (NS)' technique of Butenuth [22], which models a connected set of cells as a network of active contours (known as "snakes") linked at junction nodes or ending at termination points.

While the energy minimisation process central to snakes allows them to track slow-moving contours without modification, a number of extensions are required to adapt Butenuth's original formulation to the problem at hand here. First, the network snake energy function must be modified to reflect the significant differences in the appearance of a node point and a segment of cell wall. Then, to enable fast-moving cells to be tracked, the deterministic network snake is combined with a stochastic, particle filter-based tracker. Specifically, a Markov Chain Monte Carlo (MCMC)-based algorithm [23] is used to track junction node points, to which the network snake can be anchored. If the junction points are close together, the nodes with high likelihoods may, however, 'hijack' the particles associated with the neighbouring targets resulting in coalescing of trackers. To avoid this, as proposed in [23], a Markov Random Field (MRF) based approach is used in which the particles in close proximity to neighbouring nodes are penalised by an interaction function proportional to its closeness to a neighbour target.

As cells can enter and leave the somewhat limited field of view of the CLSM as the plant grows, a Reversible Jump MCMC algorithm [24] is added to maintain a

complete description of the cell network. Finally, given the general pressure to create high-throughput techniques [25], (semi-)automatic initialisation of the snake network is highly desirable. This is achieved via a two-level watershed function, with an interactive tool allowing the user to correct any omissions or initialisation errors.

Previous work has also utilised the network structure of plant cells to provide growth estimates over time [26]. In contrast to the approach presented in this thesis which uses a combination of particle filter trackers at junctions and co-joined snakes along wall segments, in [26], Roberts et al. represented the network structure using Hermite cubics along the cell walls and the nodes positions, and captured motion information using an MCMC scheme to discover the relevant parameters of this network. The approach allowed the measurement of growth parameters, but suffered a degradation over time due to the nature of the appearance model which is a local approach based on a matching process that is incapable of handling huge network deformation and translation over time.

1.3. Overview of thesis

The remainder of this thesis is organised as follows.

Chapter 2: Literature – Segmentation and Object Tracking

A general background on various segmentation and tracking techniques along with other works related to this thesis is presented to provide the necessary fundamentals for better understanding of the techniques proposed.

Chapter 3: Hybrid approach to Tracking Network-Structured, Multiple Deformable Objects

The proposed NS-MCMC MRF-based hybrid approach for tracking network-structured, multiple deformable objects is presented in this chapter detailing the

techniques used and steps involved in the methodology. This chapter contains additional literature reviews specific to the component techniques used.

Chapter 4: Development of NS-MCMC MRF based Tracker

As a step-by-step progress towards achieving the objective of this thesis, several incremental tracking techniques were developed. These incremental techniques are based on individual component methods and their novel combinations. In this chapter, these different trackers are presented with discussion of their individual limitations which led to the development of the hybrid NS-MCMC MRF based tracker presented in the previous chapter.

Chapter 5: Evaluation of Network-Structured Multi-target Tracking Framework

A novel method of evaluating the proposed network-structured multi target tracking framework is presented in this chapter. The steps involved in the evaluation process are detailed, and supported by literature reviews specific to the performance evaluation of multi-target tracking.

Chapter 6: Results and Discussion

Using the evaluation methodology proposed in chapter 5, the developed NS-MCMC MRF-based tracking framework is evaluated for accuracy and robustness against several real and synthetic time-lapse sequences and 3D z-stack sequences of confocal images of growing *Arabidopsis thaliana* roots. Evaluation results are presented in this chapter along with a comparison between the results of the component techniques and the hybrid approach.

Chapter 7: Tracking Network-Structured, Deformable Objects in Varying Numbers - Entering and Leaving Cells

The proposed NS-MCMC MRF based hybrid tracker is extended here to track a *variable* number of deformable objects and to cope with cells entering and leaving the field of view, which is a common occurrence when the root exhibits normal growth. The detailed methodology and the related literature reviews are presented in this chapter. Finally, the developed new NS-RJMCMC MRF hybrid tracker for tracking a variable number of targets is evaluated and the evaluation results are presented in this chapter along with a comparison against the previous NS-MCMC MRF hybrid tracker for fixed number of targets to prove that the new NS-RJMCMC MRF hybrid tracker have overcome the limitations in the previous version while maintaining tracking accuracy.

Chapter 8: Conclusion and future work

This chapter concludes the thesis by summarising the proposed NS-RJMCMC MRF based hybrid technique and discussing its performance considering the goals of this thesis. Finally, unsolved problems and recommendations for future work are provided.

1.4. Contributions

There are two main contributions from this thesis as follows.

- A novel hybrid approach for tracking network-structured deformable objects in varying numbers is presented in this thesis. The hybrid technique presented is a combination of the Network Snakes (NS) technique and the Reversible Jump Markov Chain Monte Carlo (RJMCMC) based particle filters technique and uses a novel method for (semi-)automatic initialization of the network snakes based on a two-level watershed function.

- A novel method for evaluating network-structured multi target tracking is presented in this thesis. This evaluation method is used to evaluate the developed tracking framework based on the hybrid approach for accuracy and robustness using real and synthetic time-lapse sequences and 3D z-stack sequences of growing Arabidopsis roots.

The methods developed here have been implemented within a software tool which is being used at the Centre for Plant Integrative Biology to extract model parameters such as cell counts, cell boundaries and cell wall elongation rates.

This thesis investigates how these contributions were achieved.

2 Literature – Segmentation and Object Tracking

In this chapter, a brief introduction to various segmentation and tracking techniques along with other works related to this thesis is provided. This provides the fundamentals required for better understanding of the techniques proposed in this thesis. The literature specific to each individual chapter is provided in the chapters themselves.

2.1. Image Segmentation

The real goal of segmentation is to divide the image into semantically meaningful regions, but this is challenging and in practise image segmentation is the process of delineating an image into multiple homogeneous regions or segments with similar characteristics such as colour and texture [27]. Segmentation is an important step in image analysis techniques such as object detection, object recognition and tracking. Many general and application-specific segmentation algorithms have been proposed, some of the more widely used segmentation techniques, particularly in bioimage analysis, are grey-level thresholding, edge detection, watershed segmentation and active contour models; literature surrounding these approaches will be presented in this section.

2.1.1. Grey-level thresholding

Thresholding is a simple segmentation technique often applied to grey level images to obtain a binary image with only two regions: a background and a foreground. Pixels are assigned to regions using a single cut-off intensity value known as the threshold value. Multiple thresholds are used to segment the image into multiple regions of various levels [28].

Thresholding may be global or local [28] [29], as described below

2.1.1.1. Global Thresholding

With global thresholding, a single threshold value is chosen and applied over the entire image to achieve segmentation. If the intensity distribution of the foreground and the background are quite distinct in the image, a global threshold may obtain good segmentation results. But in the real world, it is more common to encounter images where the intensity distribution of the foreground objects is highly variable over the image. In this case, a local or variable threshold would produce better segmentation results.

2.1.1.2. Local Thresholding

In local thresholding, the image is divided into various sections referred to as local windows, and each section is considered locally and assigned a threshold value based upon the intensity distribution of the foreground and the background in the local window only. Local thresholding is also known as adaptive thresholding [28].

Choosing an optimum threshold value is vital; the quality of the results obtained is highly dependent on the chosen threshold value. This threshold value may be manually chosen, but this is a time consuming process for applications that involve a huge dataset, for which implementing an automatic thresholding technique is a more

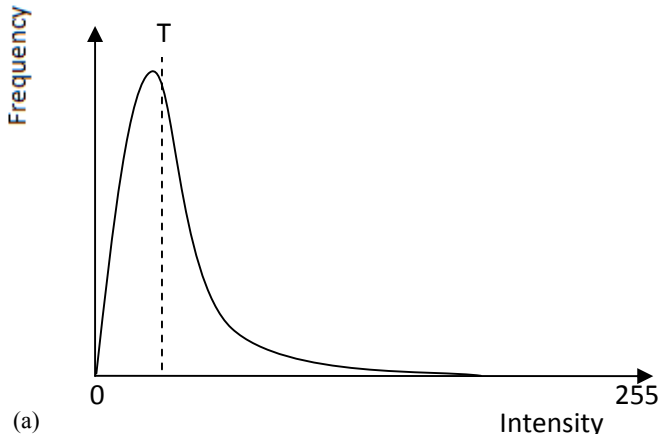
attractive option. There are several histogram-based automatic thresholding techniques available [30] to automatically choose a threshold value from the intensity histogram of the image. Based on the images considered in this thesis, and the fact that thresholding will be used as part of the junction detection, some of the relevant and more widely used histogram based thresholding techniques will be discussed here, namely the simple mean of the intensity histogram [30], Rosin's unimodal thresholding [31] and Dunn & Joseph's local histogramming technique [32].

2.1.1.3. Simple mean of the intensity histogram

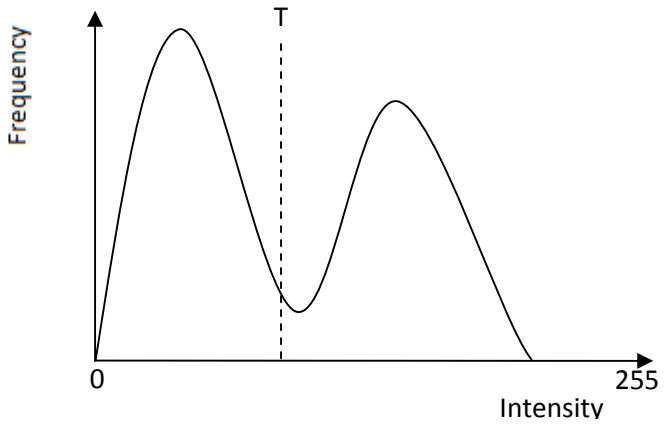
The mean of the intensity histogram can be chosen as the threshold value for segmentation of the image. If n is an intensity level and f(n) is the frequency of the corresponding intensity bin, where n ranges from 0-255, mean of the intensity histogram can be expressed as

$$\frac{\sum_{n=0}^{255} n * f(n)}{\sum_{n=0}^{255} f(n)} \tag{1}$$

This method is simple and computationally inexpensive and works well as a threshold for the images considered, where the foreground pixels are much brighter than the background pixels. This method works reasonably well for both unimodal and bimodal intensity distribution as shown in Figure 2 where the mean T lies at a happy medium for both the cases.



(a)



(b)

Figure 2. Mean of Histogram
(T- Threshold value) (a) Unimodal histogram (b) Bimodal histogram

2.1.1.4. Paul Rosin's Unimodal thresholding

Rosin describes a bi-level thresholding [31] technique which assumes that there is a dominant primary population in the image that produces one main peak located at the lower end of the histogram and a secondary population which is small enough to be ignored or sufficiently well separated from the large peak to avoid being swamped by it. Under these assumptions the histogram can be treated as unimodal, as shown in Figure 3. If necessary, the histogram can be smoothed to reduce the effect of noisy spikes. Estimating a threshold value requires the following steps.

1. Draw a straight line P-L from the peak (Point P in Figure 3) which is the highest value bin of the histogram to the high end of the histogram (Point L in Figure 3) which is the first empty bin after the last filled bin in the histogram.

2. Threshold T is then chosen such that the perpendicular distance from the frequency at T to the line P-L is the maximum.

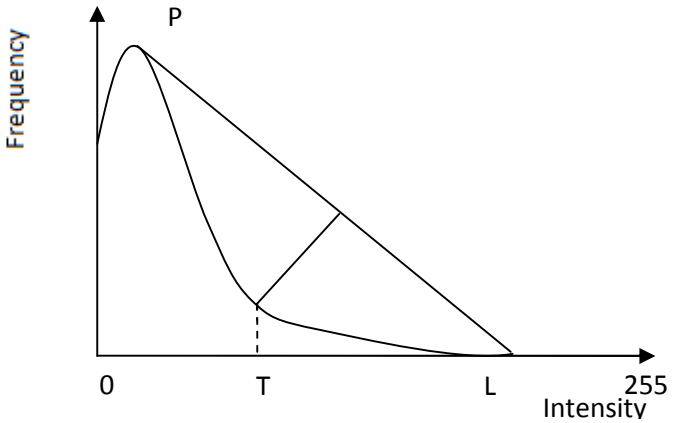


Figure 3. Paul Rosin Unimodal thresholding
T-Threshold value, L - First empty bin after the large filled bin

2.1.1.5. Dunn & Joseph's Local Histogramming technique

In Dunn & Joseph's local histogramming technique [32], it is assumed that the background noise is the dominant population in the image which forms the main peak compared to the foreground population. The threshold value is estimated from the half width measure calculated at half the peak's height (h/2) of the main peak in the histogram formed by the background of the image. The half width measure (W) is calculated by taking the average of half widths (W1 and W2 in Figure 4)

calculated at either side of the peak at half its height as shown in Figure 4. Half width W can be expressed as

$$W = \frac{W_1 + W_2}{2} \tag{2}$$

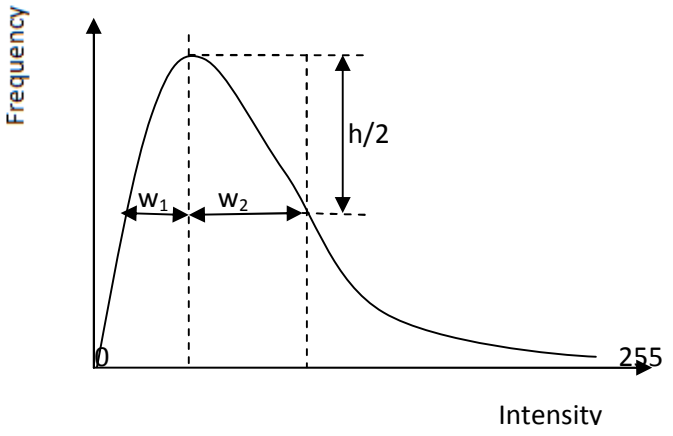


Figure 4. Dunn & Joseph local histogramming

It has been shown that the half width measure W is approximately equal to 1.2σ, where σ is the standard deviation of the dominant peak of histogram. Dunn and Joseph propose 3.6σ or 3 times the half width as an ideal measure for the threshold value T. If the peak formed by the background is more of a spike and not a Gaussian, the half width measure is too small a value. The main peak can be broadened a little by smoothing the histogram, thereby increasing the half width measure.

Thresholding is widely used for segmentation [27] [33] [34] [35]. It is computationally simpler than other segmentation techniques and is therefore attractive for applications in which a minimal computation time is a constraint.

Thresholding techniques are used in bio-imaging applications with much success [34] [35]. In [34], authors propose a connectivity based thresholding technique for segmentation of midsagittal brain MR images in which the thresholding segmentation is improved by defining region boundaries identified using Rubin's Lee path connection algorithm [35] combining the prior knowledge that can be transformed into connectivity constraints. Thresholding is also used for segmentation of cells [36] in which a global thresholding technique proposed by Otsu [37] is employed.

2.1.2. Edge detection

An edge in an image is characterised by a sharp change in the local image intensity. Edge detection is the process of identifying these sharp changes in the intensity in an image. Edge detection techniques can be classified as first derivative and second derivative techniques. First derivative techniques such as Sobel [38], Roberts [39] and Canny edge detectors [40] involve determining the intensity gradient of the pixels. Local maxima in gradient are then considered to mark edges. Second derivative techniques such as the Laplacian technique [41] seek zero-crossings in the rate of change of the intensity gradient.

2.1.2.1. Sobel Operator

The Sobel operator is shown in Figure 5. This includes two *3x3* kernels, A_x and A_y that are convolved with the image along the horizontal and the vertical direction respectively to obtain the image gradient G_x along the X direction and G_y along the Y direction [38].

$$A_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad A_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Figure 5. Sobel Operator

The gradient magnitude G and the gradient direction θ can be obtained by the following equations [38].

$$G = \sqrt{G_x^{\,2} + G_y^{\,2}} \tag{3}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{4}$$

2.1.2.2. Roberts Cross Operator

Roberts Cross operator [39] comprises of two *2x2* convolution kernels as shown in Figure 6.

$$A_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad A_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Figure 6. Roberts Cross Operator

Kernels A_x and A_y are convolved with the image along the horizontal and the vertical direction respectively to obtain the image gradient along the two diagonals, G_x and G_y. The gradient magnitude G and the gradient direction θ can be obtained from the equations (3) and (4) respectively.

2.1.2.3. Canny edge detector

In the Canny edge detector, the edge detection is achieved by the following steps [42]:

1. Smoothing the image by applying an appropriate Gaussian filter.

2. Calculating the image gradient magnitude and direction using an edge detection operator such as Roberts Cross operator.

3. Non-maximal suppression in which, if the gradient magnitude of a pixel is greater than that of its two neighbours in the gradient direction, the pixel is the local maxima and marked as an edge. Otherwise, it is marked as a background.

4. Applying Hysteresis thresholding to remove the weak edges. Two threshold values are used, one high and one low. Initially, higher threshold is applied to the image to mark all the pixels above this threshold as edge pixels, but leaving the rest as potential edge pixels. Then the edges are traced through the image using the gradient direction obtained earlier to include the faint edges that are above the lower threshold.

The Canny edge detector is an optimal detector which is based on three criteria, namely good detection, good localisation and single response to an edge [38]. But the Canny edge detector is computationally expensive [43] due to the hysteresis thresholding stage. Also, the Canny algorithm is inaccurate in detecting the branching edges [42] near junctions that are important features for the tracking application chosen in this thesis.

The Sobel edge detection operator is less sensitive to noise compared to Roberts cross and the second derivative edge detection techniques. Roberts cross operator

produces weaker response to genuine edges unless they are sharp because of its smaller kernel compared to the Sobel operator.

Edge-based segmentation techniques involve detection of boundary edges followed by linking up of the disjoint edges to form regions [44] [45]. In [44], a segmentation technique based on '*edge flow*' was proposed, in which region boundaries are detected by iteratively propagating the edge flow and identifying the locations where two opposite direction of flows encounter each other, indicating the presence of a boundary. The disjoint boundaries within a specified search neighbourhood are then connected to form closed contours followed by a post processing to reduce the number of regions to within a user specified value. In [45], authors use an edge-based approach for representing surfaces in 3D range data, in which the edge pixels are first detected using a second derivative edge operator and the edge pixels are then linked along the direction of the edge points to form closed contours. Edge-region-based segmentation techniques using growing edges based on a set of chosen critical points have also been proposed in [46] and used for 3D object segmentation of range images.

2.1.3. Watershed Segmentation

In watershed segmentation, the image is segmented into multiple regions called catchment basins, and these basins are delineated by watershed lines [47]. The segmentation algorithm simulates flooding of the image terrain by a single, steadily-increasing, water level. As the water reaches new basins, new regions are created as shown in Figure 7, where three regions 1, 2 and 3 are formed.

The image terrain is flooded from several minima in an image and when two or more floods start to merge, a watershed or dam is built to avoid the merging of floods. At

the end of this flooding process, many catchment basins are created, delineated by watersheds.

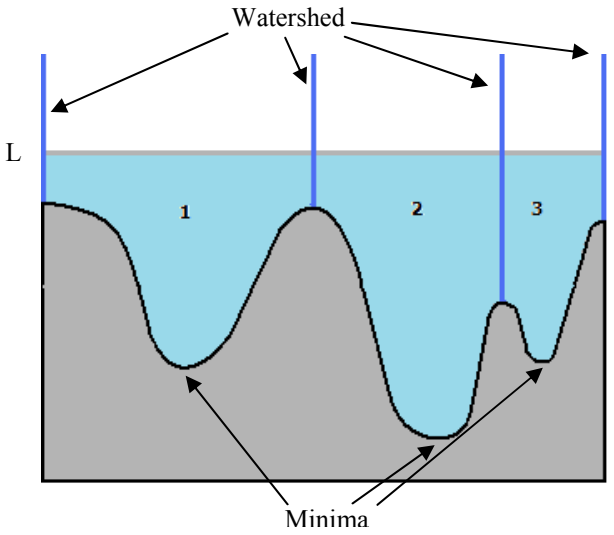


Figure 7. Watershed algorithm - Flooding
(L – Water level, the image segmented into three regions 1,2 and 3 with one minima in each)

The approach has a drawback; watershed-based techniques often lead to over-segmentation [47] [48] [49] [50]. Several techniques have been proposed to overcome this problem. In [47] and [48], a marker controlled watershed approach was proposed in which many suitable markers are provided, one for each catchment basin and the image terrain is flooded starting from every marker. Finding suitable markers is, however, an expensive and complex process which needs manual intervention. In [49], region merging was proposed to correct over-segmentation. This process examines size and shape to identify over-segmented objects, a complex task if the objects are clustered closely together. In [47], the hierarchical segmentation is used to merge the over segmented regions. The need to incorporate such measures highlights the low level nature of the watershed transform. In [51], the authors propose a new technique called *watersnakes* which represents the watershed

segmentation as an energy minimizing problem, mainly to control the smoothness of the region boundary in the segmentation result.

The watershed segmentation is used in many applications. In [52], a marker controlled watershed-based approach is used to develop an unsupervised colour image segmentation algorithm for face detection applications. Watershed segmentation is also used for segmentation of range images [53], in which an edge map is used to create input markers for morphological watershed algorithm. The watershed algorithm is another popular choice for segmentation of biomedical images [48] [49]. In [49], a hybrid approach based on watersheds and a gradient-weighted distance transform is used for segmentation of nuclei in confocal image stacks of rodent brain. Malpika et al [48] proposed an algorithm based on morphological watersheds for the segmentation of microscopic nuclei clusters in peripheral blood and bone marrow images.

2.1.4. Active contour model

An active contour model, also called a 'snake' is an energy-minimizing spline guided by internal constraint forces and influenced by image forces that pull the snake towards features such as edges [54].

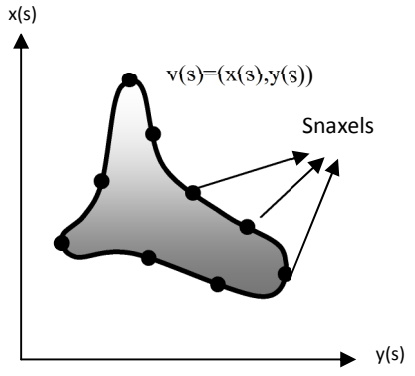


Figure 8. Snake

For a snake represented by a closed parametric curve v(s) = f(x_s,y_s) as shown in Figure 8, the combined energy equation for the conventional snake [54] is

$$E_{snake} = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{image})ds \tag{5}$$

where, E_{cont} is the continuity energy term, E_{curv} is the curvature energy term and E_{image} is the image based external energy term. $\alpha(s)$, $\beta(s)$ and $\gamma(s)$ are parameters used to control the influence of the corresponding individual energy terms. The continuity and the curvature term together form the internal energy of the snake.

If p_1, p_2, p_3... p_N are the snake points on the snake, also referred to as 'snaxels' [55] [56] and d_{avg} is the average distance between the snake points, the individual energy terms can be explained as follows [57].

2.1.4.1. Continuity term

The continuity energy term is used to maintain all the snake points at equal distances while minimizing the energy of the snake.

The continuity term at a snake point p_i is

$$E_{cont} = (d_{avg} - \|p_i - p_{i-1}\|)^2 \tag{6}$$

2.1.4.2. Curvature term

The curvature term is used to maintain the smoothness of the contour while minimizing the energy of the snake.

The curvature term at a snake point p_i is expressed as

$$E_{curv} = \left\| p_{i-1} - 2p_i + p_{i+1} \right\|^2 \tag{7}$$

2.1.4.3. External image energy term

The external energy term is the image based energy which causes the snake to be attracted towards image features like lines and edges during the energy minimisation process. The external energy term at a snake point p_i can either be the inverted values of the direct image intensity measure expressed as

$$E_{image} = -I(x, y) \tag{8}$$

or the inverted values of the gradient-based energy at that point expressed as

$$E_{image} = -|\Delta I(x, y)| \tag{9}$$

where, $|\Delta I(x, y)|$ is the image gradient.

The inverted value is to make the snake be attracted to contours with large intensities or image gradients.

The most commonly used external energy is the image gradient based energy [58] [59] which is obtained by using an edge detection technique which is explained in the previous section.

Allowing a snake to minimize its total energy to settle at the energy minimum makes it deform and become attracted towards the underlying image features like edges, while maintaining the continuity and the smoothness of the contour as shown in Figure 9.

Figure 9. Energy minimisation of a snake
(a) Initialised snake (b) Energy minimised snake

The traditional active contour models have certain limitations [60] [61] [62]. They are limited to closed object boundaries and allow no topological transformations such as merging and splitting of contours. The active contour models are highly sensitive to the initialised configuration; to obtain a good convergence, the initial configuration of an active contour needs to be very close to the target image features. Several improvements to the traditional active contour models addressing these limitations have been suggested in the past [60] [61] [62] [63] [64] [65]. McInerny and Terzopoulos [60] proposed topologically adaptable snakes which can handle topological transformations such as splitting of a snake into multiple parts and merging of snakes. This is achieved by iteratively reparameterizing the deforming snake model using a superposed grid by simplicial decomposition allowing the model to flow into complex shapes and change its topology when necessary. Velasco and Marroquin [62] proposed growing snakes that can find contours with complex

topology. Growing snakes are modelled as set of connected particles and are automatically initialised. They start from places in the image with large gradient magnitude and grow following the high gradient zones forming T-junctions when they meet. These T-junctions are then refined to find the minimal energy configuration for the complete snake.

To overcome the difficulties in topological transformations of the traditional active contour model, which requires a parametric representation, the implicit representation of contours using level set methods was proposed for example in [66] [67] and has been used for segmentation of cells [68] [69]. The level set method was first proposed by Osher and Sethian [70]. The basic idea is to represent the evolving contour (in 2D) or a surface (in 3D) as a signed distance function where the zero-level is the actual contour. The contour is allowed to move in the direction of its local normals using a defined speed function. This formulation allows automatic topological changes such as splitting and merging of curves. However, this topological flexibility of the technique also makes it unsuitable for segmentation of clustered or connected cells as discussed in [58] [69] and [71].

To improve the convergence properties of the traditional parametric active contours, Neuenschwander [63] proposed ziplock snakes which greatly simplifies the initialisation process of the contours and yields excellent convergence properties. In ziplock snakes, only the distant end points of the contour need to be defined rather than a complete polygonal approximation as required by the traditional active contour model. An initial state of the snake is then derived by progressing from these end points towards the centre taking the image information into account. In [64], the authors present a new external force called gradient vector flow (GVF) for the active contour model to address the problems associated with the initialisation and poor convergence to boundary concavities in the traditional active contour model. GVF is

a vector flow field computed from extending the gradient map farther away from the edges using a computation diffusion process thereby increasing the capture range of the external force fields and make the contours converge to the boundary concavities. GVF snakes are further improved by generalized GVF formulation (GGVF) [72] which includes two spatially varying weighting functions so as to decrease the undesirable smoothing effect caused by GVF diffusion near strong gradients. Cohen [65] describes a technique which considers a contour as a balloon, in which an accompanying inflating force is added to push the contour in the *absence* of image features. The contour with an inflating force passes over the week edges and stopped only by the strong edges. This prevents the contour being trapped by spurious isolated edge points and hence, the initial state of the contour need not be very close to the actual image features, yet provide better convergence.

Several improvements to the originally proposed gradient descent based optimisation technique [54] were also proposed to improve the convergence properties. Amini et al [73] have shown that the variational calculus based optimization proposed in [54] has several problems such as high sensitivity to noise and a tendency for snake points to shrink and cluster along the strongest portion of the edge because no constraints is placed on the inter-distance of the snakes. A scale space approach was proposed in [54] [74]. However, Amini et al [73] claim that a scale space approach could potentially destroy the edge information and could well be the source of new errors, and have proposed a dynamic programming method that allows addition of hard constraints to obtain the expected optimality, but using dynamic programming for optimization is computationally expensive [74]. A faster optimization algorithm called a greedy algorithm was proposed in [57] which allows the inclusion of hard constraints and runs much faster than the dynamic programming method, yet provides good convergence. This thesis uses the greedy algorithm based approach to

obtain an optimal configuration for a network of contours that are initialised using two-level watershed segmentation as explained in chapter 3.

Active contour based techniques have been used for the segmentation of images showing numbers of individual cells [58] [59]. A contour based technique has also been proposed [75] for the automatic segmentation of 3D confocal images of cellular features.

The implicit tracking ability of active contour models can be used for tracking applications [55] [58] [59] [76]. In these implementations, segmentation is achieved by allowing a suitably initialised snake to find its energy minimum on a single object. To track the object over a sequence, this energy minimised configuration is used as the initialisation for the next frame and is allowed to find its energy minimum to fit the underlying image data. Following this process over a sequence leads to a simple tracking.

Implicit tracking ability of active contour models is, however, limited to slow moving contours; if the movement of contours between frames is large enough that the initial energy-minimised configuration of the snakes at the previous time step becomes invalid, and located far from the underlying image features at the current time, the implicit tracking eventually fails. To cope with large movements, a mechanism to estimate the motion of contours and to make a posterior prediction of contours is necessary [77]. These are called tracking algorithms, of which there are many types [78]. Object tracking is explained in the next section.

2.2. Object Tracking

Tracking is the process of estimating one or more objects' states in a given sequence of images over a period of time. Moving objects undergo a change in state which is observed initially based on which the objects' evolving state is predicted over the sequence.

A tracking framework generally involves following two steps [78]:

1. Object detection for initialisation of the tracker – detecting the target objects in an image at time t=0 in the sequence

2. Tracking – Predicting and reacquiring the detected targets in subsequent frames over the sequence

2.2.1. Object Detection

Object detection is the process of detecting the presence and location of objects in an image. Object detection involves extraction of image features or regions of interest from the image, and the extracted features or regions are then compared against a predefined representation of the target object called an appearance model. Detection can be based on local or global image information. The most common local approach is to slide a window across an image at various scales and to identify the presence of an object using a classifier based on the extracted local image information [79] [80]. Local region based approaches can fail when the local information is insufficient for classification. Using the global information from an image provides additional clues such as global image texture and global shape context, which when combined with the local detection techniques improve the accuracy of the detection [81] [82].

Figure 10. Human Detection using template representation
(a) Circle placed over a rectangle (b) Silhouette

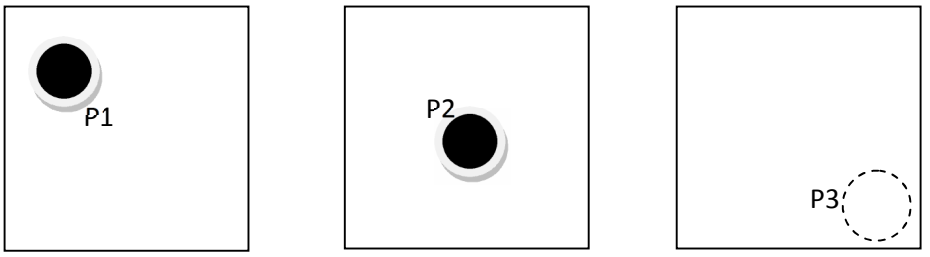
An appearance model describes the appearance of the target object in terms of parameters like shape, colour or texture, etc. For example, a circle placed over a rectangle as in Figure 10(a), or a complex silhouette as in Figure 10(b), are example representations of the shape of a human. Using the appearance model, the target object is detected using various techniques such as template matching, in which a silhouette template such as in Figure 10(b) is fitted at several candidate locations on an image and a measure of quality of fit is determined which, when exceeding a threshold measure, can be used to indicate the presence of a human.

Each detected target at time t=0 is assigned a tracker which tracks the state of each target over time.

2.2.2. Tracking

Tracking is the process of estimating one or more detected objects' trajectories over a period of time. There are three key components to any tracking framework; an appearance model that represents the target object in terms of parameters like shape, colour or texture, etc as explained above, a dynamic model or motion model that describes the dynamics of the target object and the engine that applies the appearance and motion models to perform tracking such as the Kalman filter [83] or a Particle filter [84].

A motion or a dynamic model expressed as $P(X_t \mid X_{t-1}, X_{t-2}, \ldots)$ estimates the location of an object using its prior motion. Consider the example shown in Figure 11 where an object exhibits a linear and uniform motion; an object at position P1 at time *t*-2 has moved to position P2 at time t-1. With this prior information built in to the motion model, given the previous state X_{t-1} at time *t*-1, the location of the object for the current state X_t at time *t*+1is predicted. For a stochastic process where a state X_t of a target is represented by a set of particles, the dynamic model is used to propagate the particles forward in time and the predicted location of the object is estimated by measuring the likelihood of all the particles in the state and choosing either the particle that shows a high likelihood or a weighted average of all the random positions. Any error in the motion model is then corrected from this estimation. The appearance model of the target object is used to measure the likelihood at candidate points.



(a) t-2 (b) t-1 (c) t

Figure 11. Dynamic Model

The above is only an example of a simple motion model, but there are many possible motion models for various applications [85]. In [85] and [86], multiple motion models were defined with automatic switching between appropriate motion models. In [87] [88], the targets motion model is assumed to be Markovian in which the prediction for the current state is dependent only on the previous state. Choosing an appropriate motion model plays an important role in the efficiency and accuracy of the tracking application.

The engine that applies the appearance and motion models to perform tracking are the dedicated tracking algorithms which are of many types [78]. Probabilistic tracking algorithms are more commonly used to model uncertainties. These algorithms apply Bayes' rule to evaluate the posterior probability distribution *p(x|z)* based on the prior estimates p(x) as

$$p(x \mid z) = kp(z \mid x)p(x) \tag{10}$$

Where, *k* is the normalization constant and *p(z|x)* is the measurement from the image data.

The Kalman filter [83] is one of the most widely used techniques for visual tracking. The Kalman filter is a *predictor-corrector* algorithm [89] which involves two major

steps. In the prediction stage, the previous state at *t-1* is projected forward in time to provide an estimate of the target's location at time *t*. This is known as *a priori* estimate because the estimation is only based on the prior information. In the correction stage, this *a priori* estimate is then refined by using the actual measurement at the current state to provide the refined *a posteriori* state estimate. Beymer and Konolige [90] use the Kalman filter to track people in an office corridor using disparity templates from the stereo image information for detecting people from the scene. Peterfreund [91] uses a Kalman-filter based approach for active contour models called a velocity snake model [77] for tracking of nonrigid objects in combined spatio-velocity space. In [92], a shape and size-constrained active contour model is coupled with the Kalman filter based on a constant velocity assumption for tracking rolling leukocytes *in vivo*.

The Kalman filter is often successful as long as the dynamics of the system are linear with a unimodal Gaussian distribution of likelihood density. The main disadvantage in the Kalman filter approach is its inability to cope with non-linear motion and non-Gaussian multimodal distributions [84] [24]. Attempts to solve this problem led to the development of particle filter based methods such as Condensation algorithm [84] by Isard and Blake, which approximates the posterior probability distribution as a random set of particles using factored sampling.

2.2.2.1. Particle Filters: Condensation

The CONditional DENSity propagATION (Condensation) algorithm for visual tracking is a particle filter based tracking algorithm proposed by Isard and Blake [84].

The algorithm [84] iterates over the following three steps, namely selection, prediction and measurement as explained below.

If the weighted sample set at time *t-1* is represented $\{S_{t-1}^n, \pi_{t-1}^n\}$, where *n* =1,2,..., N, S_{t-1}^i is a sample *i* and π_{t-1}^i is its likelihood weight, the new sample set at time t $\{S_t^n, \pi_t^n\}$, *n* =1,2,..., N is constructed as follows:

Selection

Select randomly sampled, uniformly distributed set of un-weighted particles $\{S_t^n\}$ by sampling N times from the weighted sample set $\{S_{t-1}^n, \pi_{t-1}^n\}$ at time *t*-1. For effective sampling from the weighted sample set $\{S_{t-1}^n, \pi_{t-1}^n\}$, the cumulative distribution of all the particles' weights is determined and the particles are selected by using binary subdivision so that samples with high likelihood weights are likely to be chosen many times.

Prediction

By applying the motion model, selected particles are propagated to new positions at time *t*. After predicting the new locations, the samples at new predicted positions have Gaussian noise added to spread the particles; in other words the particles are diffused by adding independent Brownian motion to all the particles.

Measurement

The diffused particles in the new positions are reweighted based upon the measurement of the current state density which represents the likelihood of each particle to be the object to construct the weighted sample set $\{S_t^n, \pi_t^n\}$.

The target object's tracked position is then estimated as the weighted average of all the particles in the sample set $\{S_t^n, \pi_t^n\}$.

Algorithm 1. Condensation algorithm [84]

Particle filters such as Condensation algorithm are far more effective than Kalman filters as they can deal with non-Gaussian multimodal distributions and non-linear motion.

Improved versions of the Condensation algorithm were also proposed such as mixed state condensation [85], subordinated condensation [93] and competitive condensation [94].

In mixed state condensation [85], the basic condensation algorithm is modified to allow automatic switching between multiple motion models. In this approach, multiple motion models are defined so that a discrete set of state transitions are possible at one time. All possible state transitions are maintained while the motion is ambiguous, but as soon one model predicts significantly more accurately than the others, that model will dominate. The algorithm is demonstrated by tracking a bouncing ball and tracking a natural drawing action of a hand holding a pen.

Subordinated condensation [93] is an extension to the basic condensation algorithm for tracking an arbitrary number of targets in a cluttered sequence with many overlapping objects, fully or partially occluded. In this method, a single set of particles is used to represent the state density of several objects of the same type with an explicit representation of occlusion amongst particles within the sampling framework allowing particles to occlude each other. This is implemented by introducing subordination links between pairs of particles as and when they occlude, with the occluded particle becoming subordinate to the occluder (i.e., if a particle A is subordinated to a particle B then A is behind B) and these subordination links are propagated along with the set of particles to identify occlusions. The algorithm is demonstrated by tracking birds of same species against an orange sky background in which a bird is represented by a set of 2-D points forming a skeleton of the bird.

In competitive condensation [94], the basic condensation algorithm is modified to use a competition rule to separate closely positioned or partially occluded objects while tracking a group of people. If the targets are very close to each other or partially overlapping with each other, the tracker associated with a target contains a subset of samples around a neighbour target tracked by another tracker. This overlapping subset of samples is suppressed by penalizing the samples' weights so as to reduce the effect of these samples in the state space estimate and these suppressed samples with reduced weights are less likely to be picked for the next sampling process enabling the separation of the trackers over the next frames preventing the problem of merging trackers.

Particle filters have been widely used for various tracking applications such as mobile robot localization [88] [95], space sensor surveillance [96], gestures recognition [97], people tracking [94] and in molecular cell biology [98].

Markov chain Monte Carlo (MCMC) filtering is another popular particle filter based technique used in object tracking. Particle filters often lead to the degeneracy problem [99] [100]. In [99], it has been shown that the variance of the importance weights increases over time so that after a few iterations, all but one weight will converge to zero. This is known as the degeneracy problem, which can be overcome by using a resampling scheme. The resampling scheme, however, introduces another problem [99] [100]; after resampling step, particles with high weights are selected many times which leads to a loss of particles diversity after a few iterations. This is known as sample impoverishment. MCMC algorithm can be used to solve this problem [99] [100]. Also, for interacting targets [23], the state distribution needs to be modelled as a set of joint state samples in which each joint state sample represents the state of multiple objects. For such implementations, particle filter techniques suffer from exponential complexity with the number of target objects [23]. This can

be efficiently done using MCMC MRF based algorithm [23]. The algorithm is explained in detail in section 3.5.1.

To extend the MCMC based approach to varying state dimensions, Khan et al [24] used the Reversible Jump MCMC (RJMCMC) filter to track a variable number of interacting targets. The approach was found successful when applied to real image sequences from the social insect domain. In [101], RJMCMC was used for multi-person tracking that infers the number of persons in the scene, their body and head locations and head pose. The RJMCMC MRF based particle filter technique is used in this thesis for effective tracking of varying numbers of network nodes.

Other alternatives to the above Bayesian tracking framework are:

1. To detect objects at every frame in a sequence using a suitable object detection technique and to obtain a correspondence between the detected objects at the current time *t* with the previous time *t-1* based on spatial proximity and local image features [102]. To improve the accuracy of the inter-frame objects matching, a forward-backward matching technique and motion cues may be used [102] [103]. This technique was also successfully applied for tracking of individual cells [103]. However, the method is limited to small inter-frame motion and deformation of objects where the behaviour of motion and deformation are deterministic. To model uncertainties, a stochastic model such as a particle filter based Bayesian tracking used in this thesis is more appropriate.

2. To use a temporal smoothing technique for motion detection [104]. Temporal smoothing can use past and future estimates in detecting motion at every pixel of an image in a sequence. Rather than process the incoming image sequence one frame at a time, using only the information from a short

(usually single frame) history to provide context, systems taking this approach [105] compute for each target a path that is in some sense optimal over the whole sequence. While the temporal smoothing approach can provide smooth and accurate trajectory estimates for single point targets its extension to track networks of cell walls is challenging. Fitting optimal cell wall networks to 3D (x,y,t) image data is likely to be both difficult and extremely time-consuming.

3 Hybrid Approach to Tracking Network-Structured, Multiple Deformable Objects

3.1. Overview

The proposed hybrid approach for tracking network-structured, multiple deformable objects consists of an initialised network snake data structure which is able to deform and translate over relatively large transformations by being coupled with dedicated trackers at the junction nodes. This chapter will explain in detail the steps required to achieve this. During the development of this hybrid approach, several incremental tracking techniques were developed as a stepwise refinement towards achieving an effective tracking. These different tracking techniques are presented in chapter 4, along with a summary discussion on their individual limitations, which led to the development of the NS-MCMC MRF based hybrid approach presented in this chapter and further extended to cope with varying number of targets as described in chapter 7.

In outline, the proposed NS-MCMC MRF based hybrid approach is given in Algorithm 2. The output of the algorithm is a data structure representing a network of snakes, forming contours around the segmented structures, in this case the cells. This structure can be used to calculate various parameters of the segmented objects,

such as areas and cell wall lengths between junctions, and additionally used as a foundation for further processing, such as identifying a common feature inside each cell.

1. Image acquisition
2. Initialisation of the network snakes data structure using a two-level watershed transform.
3. Segmentation (refinement of cell wall estimates) by a variant of Butenuth's network snakes employing a modified energy function at the junction nodes
4. Tracking of the cell network

   At each frame:

   4.1. Projecting network nodes forward in time using an MCMC based particle filter

   4.2. Warping the network snake from the previous frame onto the new node positions

   4.3. Refining the network description by minimising the energy of the warped network snake.

   4.4. Updating the MCMC joint state particle distribution using the refined network.

Algorithm 2. NS-MCMC MRF based hybrid technique

3.2. Image acquisition

Arabidopsis plants containing a green fluorescent protein membrane marker were grown as detailed previously [106]. Roots of 7 day old plants were imaged via CLSM using Nikon Ti Eclipse and Leica SP5 systems. 2D time lapse images of expanding roots in the elongation zone were taken using roots growing in glass-bottomed petri dishes using the Leica microscope, and on a custom-designed vertical stage using the Nikon microscope. Either a 2D sequence through one plane was captured directly, or a thin z-stack was captured and a plane through this was later chosen for processing. For some of the sequences, the frames consisted of the z-stack itself, rather than a time sequence. Sequences were stored as uncompressed TIFF or BMP images, and loaded for processing as necessary.

3.3. Initialisation of the network snake data structure

In the initialisation stage, the network snake structure must be configured to lie approximately over the network of cells. Due to the potentially large size and complex topology of the network snake structure, entirely manual initialisation is undesirable. Therefore the approach adopted here is to use a fully automatic initial pass, whose results can then be tidied up using manual interaction. Thus, initialisation is achieved as follows:

3.3.1. Segment the image using a two-level watershed based technique:

1. Noise reduction - The images are pre-processed to reduce the noise by applying an appropriate smoothing technique such as Median, Gaussian or Anisotropic smoothing to the images, depending on the image noise. Because of the presence of impulse noise, which is quite common in the images considered in this thesis, small scale median filtering (of a 3x3 neighbourhood) which is good at removing impulse noise while retaining the edges, coupled with Gaussian smoothing

(OpenCV implementation of Gaussian smoothing using a *3x3* Gaussian kernel with $\sigma = 0.5$) has been found to work well for images with high levels of noise such as the synthetic sequences considered for experimentation in section 6.3.

2. Apply a two-level watershed [107] transform

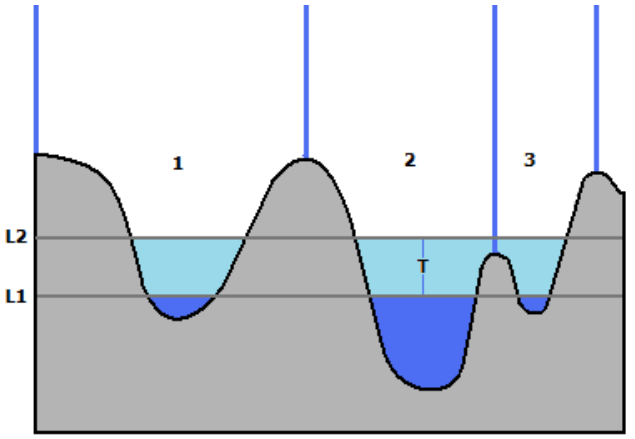


Figure 12. Two-level watershed algorithm
(L1 – Lower watershed level and L2 – Upper watershed level = L1+T)

The original watershed algorithm simulates flooding of the image terrain by a single, steadily-increasing, water level. As the water reaches new basins, new regions are created. Many such regions can be produced, leading to over-segmentation if the surface is textured or noisy. To prevent this, a second watershed level is used, above and separated from the first water level by a user defined value *T*. New regions are still only seeded by the lower water level, but are immediately flooded to the upper level. As a result, any surface features lower than *T* intensity values above the rising water are engulfed. Consider the intensity map of an image cross-section in Figure 12, where three basins are formed, of which, basin 3 is created by noise. As shown, lower watershed level *L1* forms three regions resulting from three basins, but when immediately

flooded to the upper watershed level *L2* (*L1+T*), basin *3* gets flooded and engulfed resulting in only two large regions. This reduces over-segmentation and leads to the formation of fewer, larger reported regions.

The flooding agent used in the watershed algorithm is based on a 4-way flood-fill pattern as shown in Figure 13.

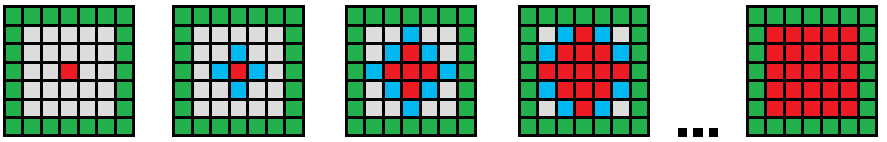


Figure 13. 4 way Flood fill pattern

An example of the two-level watershed segmented output is shown in Figure 14a.

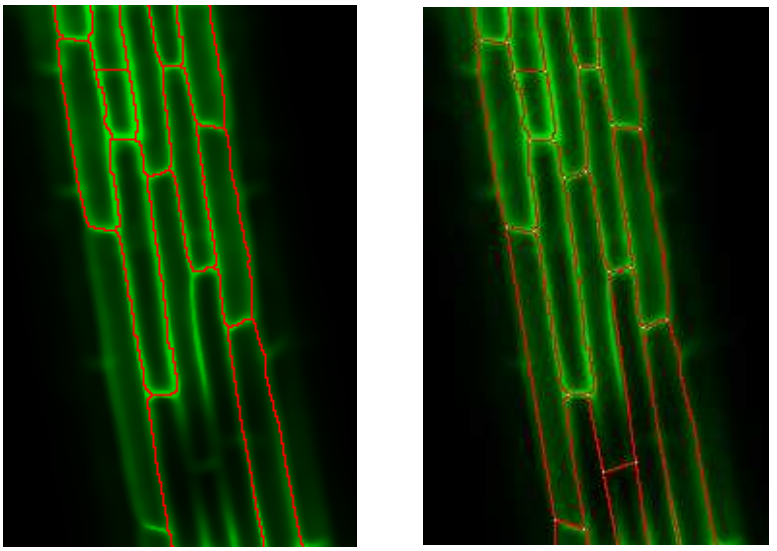
3.3.2. Detect wall junctions (node points) in the segmented output:

As the flood fill and pixel visiting pattern used in the watershed segmentation is a 4-way flood fill pattern, a junction can only branch through four possible directions: x+1, x-1, y+1 and y-1. By exploiting this, a junction detector is implemented which involves counting the 4-neighbours of all red pixels, and a pixel with more than two red neighbours is a junction (watersheds in the previous stage are marked in red). Junctions manually added after processing are inserted directly into the resultant data structure – see below for more information on user refinement of the initial automatically produced data structure.

3.3.3. Define network topology:

Snakes are initialised by walking along the boundaries between each pair of detected junctions, and placing snake points at regular intervals. Visited pixels are flagged so that they will only be included once. This process results in a network of snakes connected together at nodes as illustrated in Figure 15. This combination of node points and snakes forms our initial network configuration at time t=0.

The two-level watershed algorithm suppresses the common undesirable effect of over-segmentation, but if *T,* the difference between two watershed levels is set to avoid over-segmentation, some of the actual cells are likely to be missed (Figure 14a). These missed cell walls are manually connected with straight lines placing snake points at regular intervals. A junction picking tool allows the user to pick the missed junctions, which are then connected together with straight lines to form approximate boundaries (Figure 14b).



(a) (b)

Figure 14. Initialisation of the Network Snakes
(a) Two-level Watershed Segmented output; note the image is under- rather than over-segmented
(b) Manually completed initial segmentation.

The tool also allows spurious junctions, created by extreme noise or the CLSM sampling from too thick a slice, to be removed.

At this stage, we have an initialised network that approximates the underlying topological structure. This network is then refined by minimising the energy of the network snakes as explained in the next section.

3.4. Segmentation using Network Snakes

3.4.1. Network Snakes

The *network* snakes methodology was first proposed and used for segmentation of satellite imagery and microscopic cell imagery [22], though no time-based processing was attempted. Network snakes aim to expand the applicability of the original snakes method by connecting several independent open ended snakes together to form a global network. This allows neighbouring snakes to influence each other through their common nodes. The network snakes methodology is an appropriate technique for situations in which objects are delineated by shared object boundaries, and where deformation in the structure of the boundaries happens on a global scale rather than being entirely local as is the case in the images of cell growth.

We will now take a more formal look at the network snakes architecture. Figure 15 demonstrates the constituent elements, namely terminating nodes (white with a black spot), snake boundary elements (black) and junction node points (white).

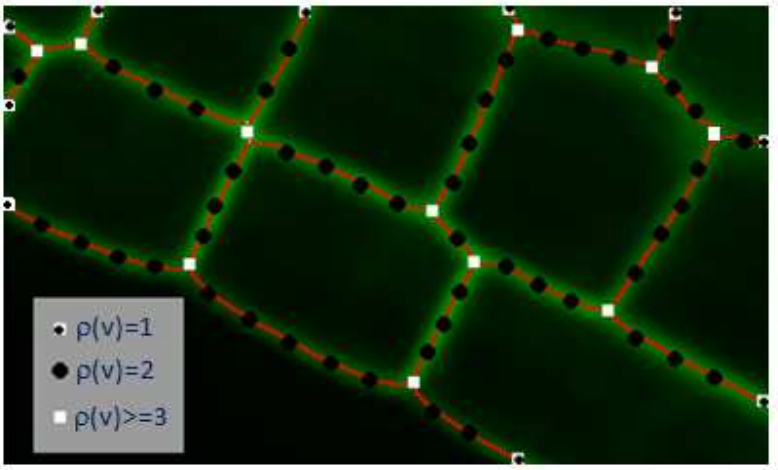


Figure 15. Schematic showing the network snakes representation of a connected set of Arabidopsis cells, as seen in CLSM imagery.
Each contour comprises a number of internal linked points (connectivity 2), with contours linked at junction nodes (connectivity >=3) and forming termination points (connectivity 1) where they cross the image boundary.

The ends of the open ended snakes are therefore generally referred to as nodes, of which there are two types. Nodes can be:

1. on the image boundary, $\rho(v)=1$

2. a junction joining 3 or more snakes, $\rho(v)>=3$

where $\rho(v)$ denotes the degree of nodes

The internal energy term of the traditional active contour needs to be altered for two special conditions to reflect the different types of nodes involved in network snakes. First, the end points on the image boundary ($\rho(v)=1$) are restricted to move only along the boundary and include only the external energy term and a modified curvature term as shown in equation (11). The continuity term cannot be used at boundary end points due to the lack of data outside the image. If n is a node point on the image boundary, then the modified curvature energy [22] at n is:

$$\beta(V_n - V_{n-1}) - \beta(V_{n-1} - V_{n-2}) \quad (11)$$

Second, for junctions joining three or more snakes, the continuity term cannot be used as no control can be maintained over continuity at a junction point. Only a modified version of the curvature term can be used to maintain the shape of each contour separately. Assuming v_a, v_b and v_c represent three contours joined together at a common nodal point *n*, the total internal energy at the common nodal point *n* can be expressed [22] as

$$\begin{aligned}&\beta(v_{a_n} - v_{a_{n-1}}) - \beta(v_{a_{n-1}} - v_{a_{n-2}}) + \\ &\beta(v_{b_n} - v_{b_{n-1}}) - \beta(v_{b_{n-1}} - v_{b_{n-2}}) + \\ &\beta(v_{c_n} - v_{c_{n-1}}) - \beta(v_{c_{n-1}} - v_{c_{n-2}})\end{aligned} \tag{12}$$

In Butenuth's formulation, the external energy term is in all cases the image gradient, which is calculated using a gradient based edge detector such as the Sobel operator [38]. In this work, given the nature of the confocal images explicitly marking edges (ie. cell walls), a simple direct measure of image intensity has been found to work well; to support other applications, however, the user interface allows a choice between both options. However, a modification of the external energy for the special case of junction nodes has proved necessary, and is proposed in the next section.

3.4.2. A new appearance-based energy for junction nodes

Basing external energy at each snake point on some measure of image gradient or intensity level is generally a sensible approach, and allows comparison of network- and traditional- snake architectures. The edge detection methods used to estimate gradient, however, assume the presence of a step edge in intensity; this is valid along the majority of a snake but, crucially, not at the junction nodes (which is why the Canny operator fails at similar locations). Figure 16 shows close up views of various junctions that do not produce a large response from a standard edge detector. The

output of the Sobel and similar edge detection operators is therefore unpredictable at junctions, where their central assumption of linear patches of opposing contrast is violated. Also, there is no specific increase in the intensity levels at junctions over that at the walls; so even a direct intensity measure may not produce a sharp response. An alternative way to represent a junction point is to use a ridge detection technique [108] to detect the medial axis of the cell boundaries that form the junction point. Due to the high foreground noise and lack of clear boundary information at the junctions, the detected ridges will be fragmented. Thus, the energy minimisation process may not place the nodes correctly if a gradient based energy, a ridge based energy or a direct intensity measure is used, and this in turn may affect the positions of the connected snakes. In general, care must be taken that the energy functions used at nodes and snake elements are appropriate; it should not be assumed that they will be identical.

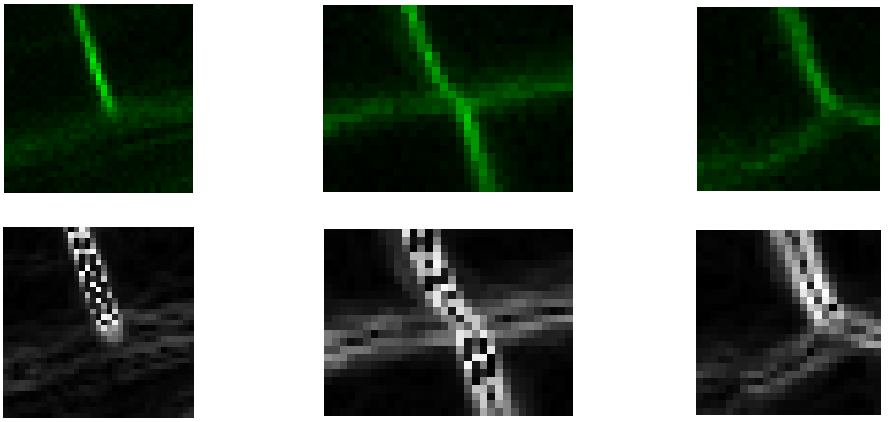


Figure 16. Close up of junctions and corresponding output from Sobel edge detector (Bottom row), illustrating why a step edge detector will not produce a large response at the junction centre. Note the lack of areas of opposing contrast at the centres of the junctions. Indeed, there is not even a peak in intensity, so even a direct intensity measure will not produce a specific response for a junction.

To overcome this problem in the current application, an appearance-based external energy is used at junctions. For this, an appearance model has been developed specifically to determine the appropriate energy weights for junction node points.

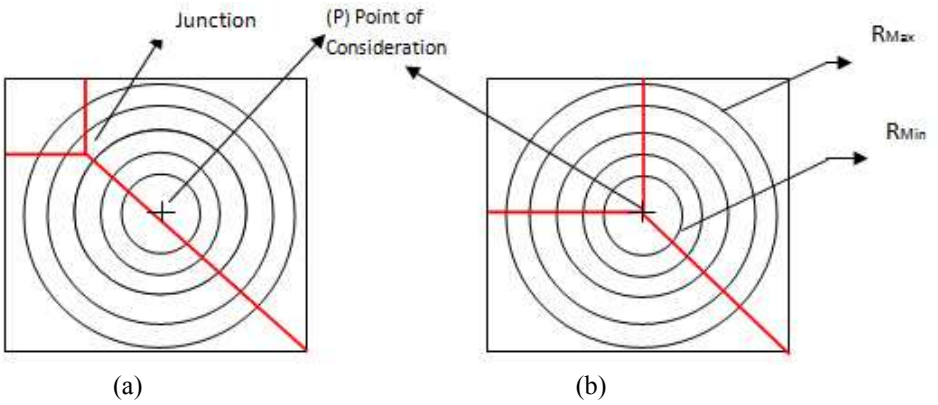


Figure 17. Appearance model used to determine likelihood factor for candidate point to be on a junction node.
(a) Candidate point off the junction, $W=2^3=8$
(b) Candidate point on the junction, $W=5^3=125$

Defining an appearance model for a node is a challenging task, especially if the nodes are very close to each other. An approach based on examining concentric circles to see how many edges they intercept is proposed as follows. Consider Figure 17, which shows a node formed at the junction of three red lines. Let the point under examination be P and the radius of the image region taken into account when assessing P be R_{Max}. To obtain a weighting factor L reflecting the likelihood that P is a junction,

1. Initialise the weighting factor L = 0 and R = R_{Max}

2. Along the circumference of the circle with radius R and centre P, if a steep increase in intensity level (above a threshold value calculated using a local thresholding technique described later in this section) is encountered, it is counted as an intersection with the cell wall. Count the number of lines or cell walls intersecting the circumference of the circle. If the count is more than two, increment the weighting factor L by 1.

3. Decrease R by 1 and go to Step 2 if R >= R_{Min},

4. To increase the difference between the low and high weighting factors, the resultant weight W is chosen as a proportional function of the weighting factor L such as a negative log function or a cubed value. In this thesis, resultant weight W is chosen as a cubed value L^3.

Consider the example shown in Figure 17a, in which the candidate point is at a distance from the node point. This will receive a low weight compared to the example in Figure 17b, where the candidate point coincides with the junction.

Choosing appropriate values for R_{Max} and R_{Min} is important to prevent enclosing neighbouring nodes in the search, or missing the desired node. Choosing a large value of R_{Max} might include boundaries of its neighbour node and falsely increase the likelihood value whereas choosing a small value provides insufficient information for the appearance model and it may miss the junction entirely. Prior information from the network structure is used to initialise the R_{Max} as shown in Figure 18 .

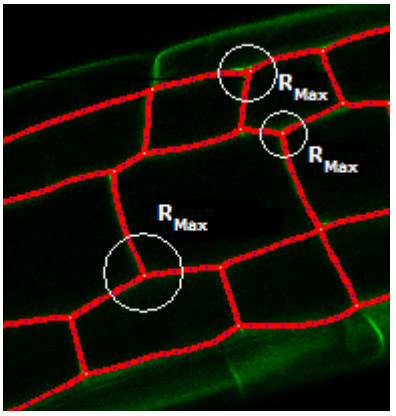


Figure 18. Choosing R_{Max} from the Network.
(R_{Max} is chosen as half the distance between the node and its nearest neighbour node)

For a given node R_{Max} is chosen as half the distance between that node and its nearest neighbour node in the network. R_{Min} is set as the average thickness of the cell walls, which normally varies between 3 to 5 pixels in the datasets considered.

Therefore, the modified energy term at the junction nodes joining three or more snakes, taking into account, external energy becomes:

$$\begin{aligned}&\beta(V_{a_n} - V_{a_{n-1}}) - \beta(V_{a_{n-1}} - V_{a_{n-2}}) + \\ &\beta(V_{b_n} - V_{b_{n-1}}) - \beta(V_{b_{n-1}} - V_{b_{n-2}}) + \\ &\beta(V_{c_n} - V_{c_{n-1}}) - \beta(V_{c_{n-1}} - V_{c_{n-2}}) + \gamma W\end{aligned} \tag{13}$$

where, W is the new appearance-based energy obtained from the described junction appearance model.

After initialising the network snakes and defining the network topology, the final segmentation output is achieved by allowing the network snakes to minimize their energy until the contours reach an energy minimum, which is a trade-off between attraction to cell boundary features, the network topology and the physical constraints embedded in the snake model. The energy terms for snake points, node points joining three or more cell walls and node points on the image boundary are given in equations (5), (13) and (11) respectively.

3.4.2.1. Choosing a threshold value for the local image region

In the implementation of the junction detector appearance model presented above, a threshold is applied to segment the local window around a junction to extract the foreground cell walls in the local window which are processed further to determine a likelihood for a candidate junction point. For this purpose, a suitable and effective automatic thresholding technique needs to be chosen to automatically determine a suitable threshold value. Based on the images considered in this thesis, some of the widely used histogram based thresholding techniques will be considered here, namely Rosin's unimodal thresholding [31] technique, Dunn & Joseph's local histogramming [32] technique and the simple mean of the intensity histogram [30].

In what follows, the results of using these techniques in the junction detector model applied to the real image sections of confocal images of *Arabidopsis thaliana* root are presented and discussed, leading to a conclusion justifying the chosen thresholding technique.

Figure 19 shows some image sections around junction points obtained from the confocal images of Arabidopsis roots and the corresponding intensity histograms of the green channel of the sections. The following histogram characteristics are observed, and influence the method used to choose a threshold value.

1. Bright pixels in the image belong to the foreground and the dark pixels belong to the background

2. The histogram of less-noisy regions appears to be more of a unimodal distribution (See Figure 19, H1 and H2)

3. The highest bin which is the main peak of the histogram is always formed by the background pixels

4. The intensity distribution ranges from 0-255 with few foreground pixels towards the bright end of the histogram

5. The histogram of noisier regions in the image produces more of a bimodal distribution (See Figure 19, H3 and H4(a)) which is clearly visible in the smoothed version (See Figure 19, S4(b) and H4(b)) of the image section in Figure 19, S4(a).

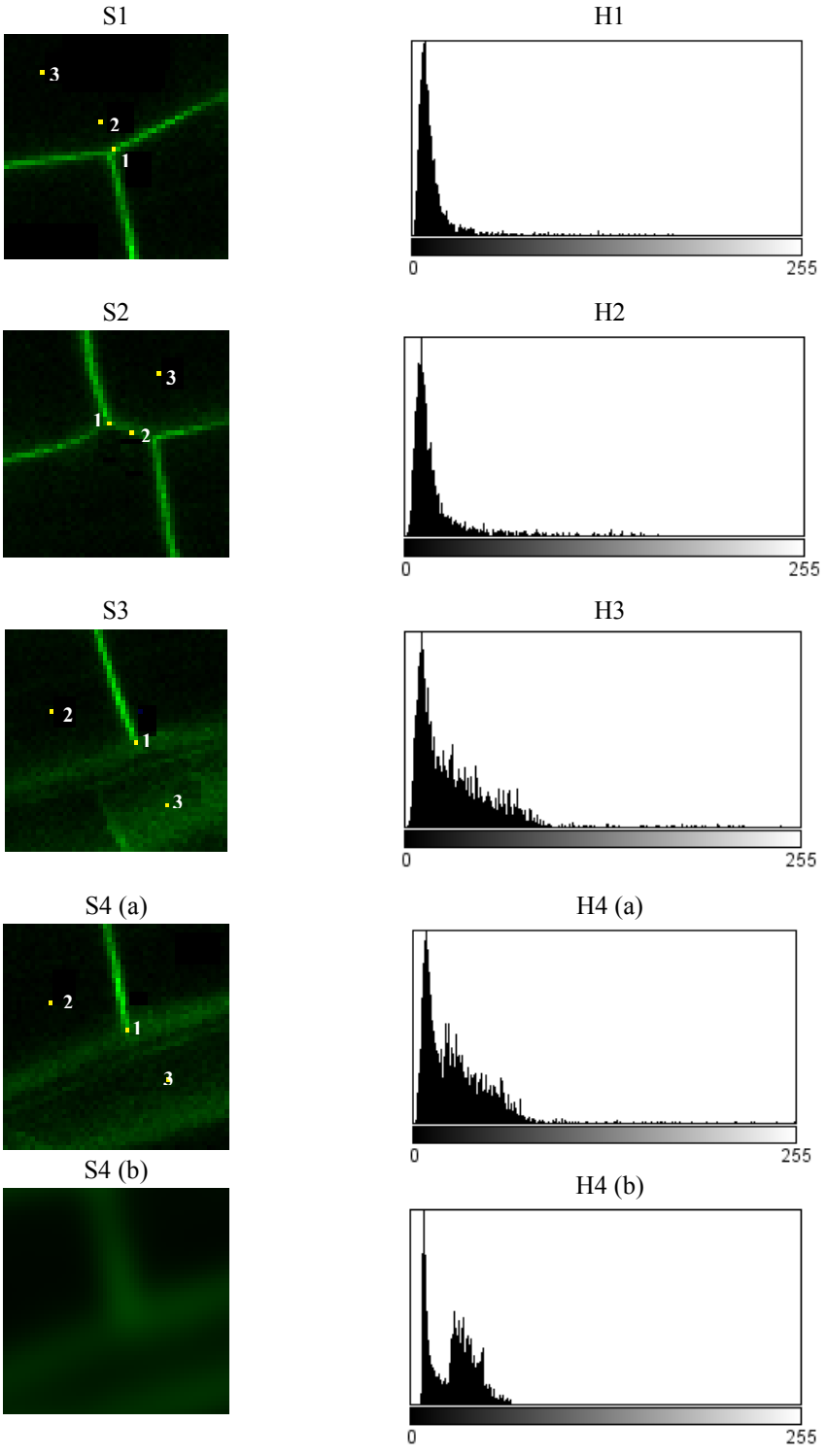


Figure 19. Sample Image sections and its histograms.
Sections S1 and S2 are less noisy compared to sections S3 and S4 (a). S4 (b) is the smoothed version of the S4 (a) and H4 (b) is the intensity histogram of the section S4 (b) (Numbered points are the points where the junction detector is evaluated; the corresponding likelihood weights are tabulated in Table 3)

The representative image sections S1-S4 in Figure 19 were carefully chosen to consider noisy to less noisy regions and the candidate points include junctions and non junction points. For all the image sections S1-S4 in Figure 19, the junction detector model is applied for the junction points and non-junction points using the three different thresholding techniques mentioned above, and the likelihood weights obtained are tabulated in Table 1. In all cases, candidate point *1* is the junction point which is expected to show a high likelihood weight and candidate points *2* and *3* are randomly chosen non-junction points that are expected to show a lower likelihood weight.

Image Section	**Paul Rosin**			**Dunn & Joseph**			**Mean of Intensity Histogram**		
#	***1***	***2***	***3***	***1***	***2***	***3***	***1***	***2***	***3***
S1	1000	1	0	1000	1	1	1000	27	0
S2	2197	729	0	1728	512	1	1728	729	0
S3	64	0	0	1	1	1	1728	0	1
S4	1	0	0	64	0	1	1000	0	0

Table 1. Likelihood weights from the junction detector using different thresholding techniques for candidate points 1,2 & 3 marked on the image sections in Figure 19. Error highlighted. In all cases, candidate point 1 is the junction point which is expected to show a high likelihood weight. Note that using Rosin's and Dunn & Joseph's methods produced a lower likelihood weight for junction points for noisy image sections S3 and S4, but using the 'mean of intensity histogram' method produced a high likelihood weight as desired.

Image Section	Paul Rosin	Dunn & Joseph	Mean of Intensity Histogram
S1	33	16	17
S2	30	20	22
S3	50	20	32
S4	75	20	28

Table 2. Threshold values obtained using different thresholding techniques for image sections in Figure 19

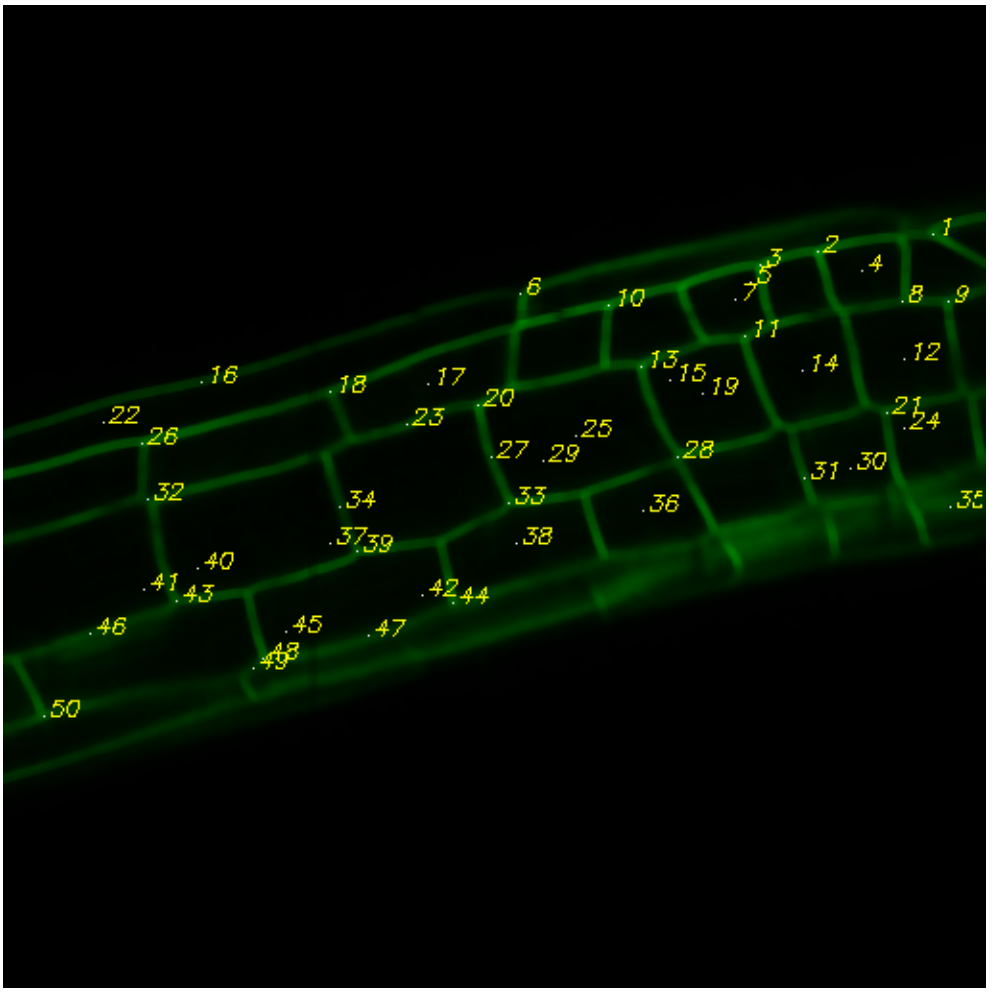
Results show that for the noisy image sections S3 and S4, where the junction detector model applied at candidate junction point *1* is expected to show a high likelihood weight, in fact using Rosin's and Dunn & Joseph's thresholding techniques, the model returned a very low likelihood weight. In all cases, using the 'mean of the intensity histogram' method to set a threshold, the junction detector model performed consistently well and produced expected likelihood weights for junction and non-junction points.

The reason for the failure of the Rosin's and Dunn & Joseph's techniques for the images considered in this thesis is as follows. Using three different techniques, threshold values are calculated for the image sections in Figure 19 and are tabulated in Table 3. Dunn & Joseph's local histogramming method consistently returned a low threshold value for all the sections, which performed well for some of the image sections but failed on others. The peak formed by the background is very steep and is more of a spike rather than a Gaussian and hence, the half-width of the peak is too low a measure resulting in a low threshold value. This is not a good measure for thresholding the images considered here because a segmentation with a very low threshold value will include a lot of background noise.

On the other hand, the threshold value obtained from Rosin's Unimodal thresholding method is always greater than that produced by Dunn & Joseph's local histogramming method and the mean of the intensity histogram. Choosing a high threshold value for segmentation is more likely to lose many foreground pixels. In Rosin's method, a few bright foreground pixels located towards the bright end of the histogram pull 'the first empty bin after the last filled bin' (point L in Figure 3) towards the bright end resulting in a very high threshold value. Using too high a threshold is not good for thresholding the images considered as this will omit many foreground pixels in the segmentation results.

As observed in Figure 19, the noisier regions in an image show a bimodal distribution while the less-noisy regions show a unimodal distribution. The mean of the intensity histogram is expected to work reasonably well on both unimodal and bimodal distributions. The results show that using the mean of the intensity histogram method, the junction detector model performed consistently well and produced the expected likelihood weights for all the candidate points in the four image sections considered, compared to Rosin's method and Dunn & Joseph's method.

Also, using the simple mean method for the appearance model has provided true likelihood weights at all the junction and non-junction points in the images considered in this thesis as shown in an empirical result with 50 candidate points picked at random positions from an image including junction and non-junction points provided in Figure 20.



p	W	p	W	p	W
1	1331.00	**18**	1000.00	**35**	8.00
2	2197.00	**19**	0.00	**36**	0.00
3	2197.00	**20**	1331.00	**37**	27.00
4	0.00	**21**	2197.00	**38**	0.00
5	64.00	**22**	64.00	**39**	1728.00
6	343.00	**23**	1.00	**40**	0.00
7	27.00	**24**	27.00	**41**	0.00
8	1331.00	**25**	64.00	**42**	0.00
9	2197.00	**26**	2197.00	**43**	1000.00
10	729.00	**27**	1.00	**44**	729.00
11	1331.00	**28**	1728.00	**45**	0.00
12	1.00	**29**	0.00	**46**	1.00
13	1331.00	**30**	0.00	**47**	1.00
14	8.00	**31**	1.00	**48**	1331.00
15	8.00	**32**	1728.00	**49**	343.00
16	1.00	**33**	1728.00	**50**	729.00
17	1.00	**34**	1.00		

Figure 20. An image marked with candidate points ***p*** (numbered) and the corresponding likelihood weights ***W***

Based on these results, the simple mean of the intensity histogram method has been chosen to automatically find a threshold for the junction detector appearance model presented earlier in this section which has proved successful as seen from the results presented in chapter 6.

3.5. Tracking cell networks

As important as identifying the cell boundary is demarcation of the cell as an entity, and maintenance of this information over time. Once an energy-minimised representation of the cell network has been created, it can be used to track movement and deformation with the help of particle filter tracking, as described below.

3.5.1. Tracking network nodes using an MCMC-MRF particle filter.

A Markov chain Monte Carlo - Markov Random Field (MCMC-MRF)-based tracking approach for interacting targets [23] is used for tracking the network *junction nodes* at each frame. In an MCMC-based technique, a traditional particle filter's importance re-sampling step is replaced by MCMC sampling.

3.5.1.1. MCMC-based particle filter

The MCMC-based particle filter algorithm iterates over the following steps [23]:

1. At time *t*-1, the targets' state is represented by a set of *N* un-weighted samples X_{t-1}. Each sample is a **joint state** sample which represents the complete state of all *n* targets.
2. Initialize the MCMC sampler by selecting a random joint state sample which is the reference sample $X_t^{'}$ from the set of *N* un-weighted samples X_{t-1} at time *t*-1, apply a motion model to all the targets and calculate the likelihood P for *n*

targets at the new positions after applying the motion model.

3. Perform Metropolis Hastings iterations to obtain *M* joint state samples. Discard the first *B* samples for burn-in and obtain the remaining *N* samples X_t which represent the targets' state at time *t*.

   a. From a set of *N* un-weighted samples X_{t-1} at time *t*-1, randomly select a joint state sample X^r_{t-1}

   b. Randomly choose a target *i* from *n* targets and apply the motion model

   c. Apply Gaussian noise to the selected target *i*'s location

   d. Calculate the likelihood probability P' for the target at the new position and calculate the acceptance ratio A. As shown in [23], by considering one target at a time, most of the factors in the acceptance ratio cancel and the simplified acceptance ratio term A is

$$A = \frac{P'_i}{P_i} \tag{14}$$

   where,

   P'i is the likelihood probability for the proposed state of the target i

   Pi is the likelihood probability for the current state of the target i

   e. if A >= 1, update the i^{th} target's position of the initialized reference sample X_t' with this new position with probability P'. Otherwise, also update the target's position with probability A.

   f. Ignore first B samples for burn in, and thereafter, add a copy of X_t' to the current set X_t.

4. At the end of the Metropolis Hastings iteration, a sample set X_t with *N* joint state samples are returned which represent the targets estimated state at time *t*. The node locations are then estimated by taking a weighted average of all the particles in the tracker's state. This location is visualised in all figures.

Algorithm 3. MCMC-based particle filter algorithm

For the acceptance ratio in equation (14), the likelihood probability is obtained by evaluating the measurement as a Gaussian density function [109]:

$$P(Z_t \mid X_t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{15}$$

The Metropolis Hastings algorithm [110] is used in this thesis to simulate random walking. Several MCMC sampling algorithms have been proposed in the past which are described and compared by Tierney [111]. The Metropolis Hastings algorithm is widely used in computer vision, and in particular has been successfully used on similar tracking applications [23] [112].

The Metropolis Hastings algorithm is demonstrated in Figure 21. The particles p1 and p2 are allowed to make random steps to eventually reach the positions p1' and p2' respectively (particle p1 makes steps 1, 2, 3, 4 and 5 to reach p1' and particle p2 make steps 1, 2, 3 and 4 to reach p2'). After making a random step, the quality of fit or the likelihood of the particle in this new position is measured and the acceptance ratio is calculated from equation (14) which is used to decide whether to accept or reject a step.

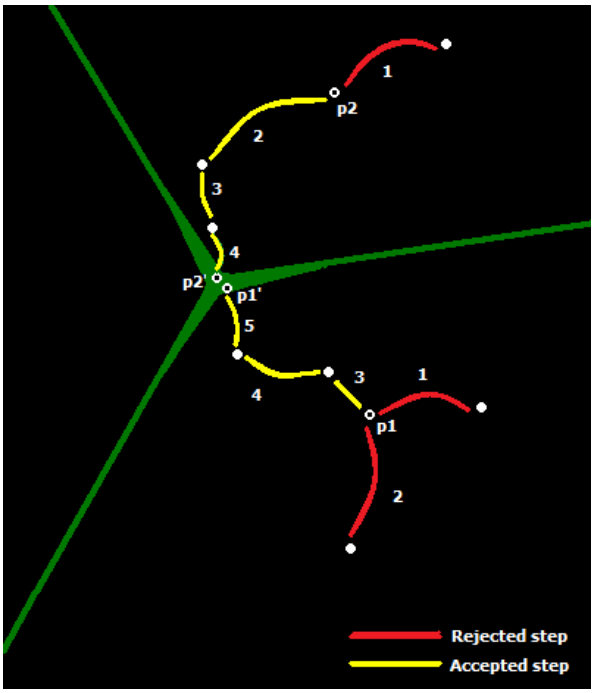


Figure 21. MCMC – Random walking.
Particles p1 and p2 make random steps (in-order as numbered against each step), eventually reaching a node.

In Figure 21, the random steps which are marked in red are rejected steps and those marked in yellow are accepted steps. If the particle gets closer to a junction node because of a random step, the likelihood of the particle will probably be higher for this new position than for the previous position making the acceptance ratio $A>1$ resulting in the step being accepted and on the other hand, if the likelihood gets poorer, the step is accepted with a probability A to prevent the particle getting caught at a local maxima. Though each random walk is aimed at improving the likelihood of the particle to eventually settle the particle at a best possible location at the end of the iteration, particles at low likelihood locations are also accepted with a finite probability during the iteration to maintain the spread of the particles.

Normalised value (between *0* and *1*) of the likelihood weight obtained using the appearance model presented in the previous section is used as the quality of fit of the candidate particles to a junction node.

3.5.1.2. MRF based approach for interacting targets

While the tracker is used to help maintain the network over time, there are some problems, which are associated with tracking in general. Coalescence is a common problem in multi-target tracking. When two targets tracked by distinct trackers are in close proximity, both trackers tend to become associated with the target with the highest observation probability. In the current context, if the target nodes are close together, the nodes with high likelihoods may 'hijack' the particles associated with neighbouring targets. Over a short number of time steps, this tends to cause the tracker to migrate over to the higher quality target. To avoid this, as proposed in [23], particles in close proximity to neighbouring nodes are penalised. This reflects the physical fact that two junctions cannot occur in the same place at the same time in the datasets considered in this thesis, as shown in the distribution of distances from junctions to their nearest neighbours in Figure 22. Only neighbours within a 25 pixel radius are shown in the distribution. Ground truth data of 60 images is used to plot the distribution which shows that over 98% of the 1210 junctions considered are at least 5 pixels away from their neighbours.

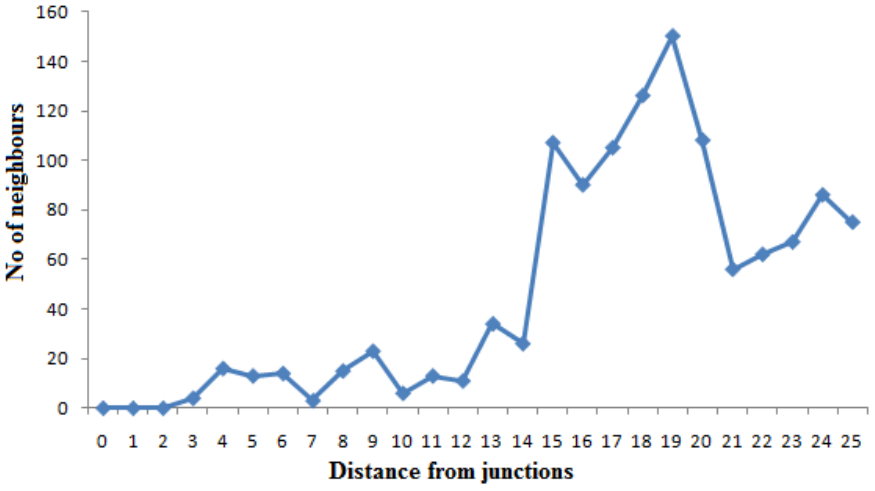


Figure 22. Distribution of distances from junctions to their nearest neighbours.

To achieve this, a Markov Random Field (MRF) is constructed at each time step, forming a graph of targets that are in close proximity as shown in Figure 23, where interacting nodes are connected with a red line. Then, the likelihood of particles is penalised by an interaction function proportional to its closeness to a neighbour target [23]. The interaction function for target m with neighbour node n at time t, $\psi(X_{mt}, X_{nt})$ is an exponential Gibbs function:

$$\psi(X_{mt}, X_{nt}) = \exp(-f(X_{mt}, X_{nt})) \tag{16}$$

where $f(X_{mt}, X_{nt})$ is a penalty function between the interacting targets which is inversely proportional to the distance between the interacting targets (d_{mn}).

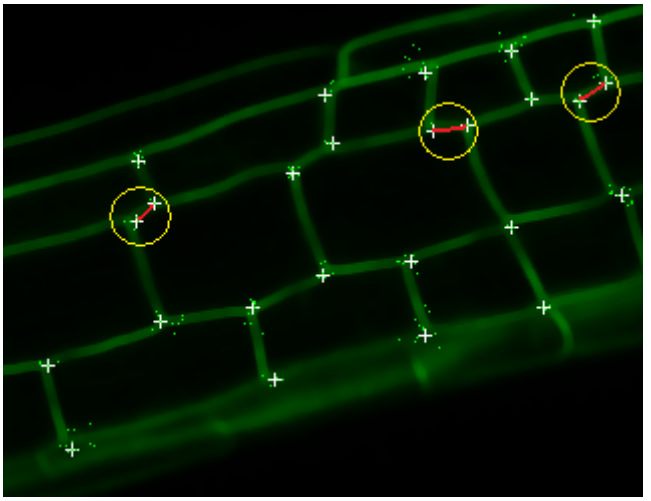


Figure 23. Markov Random Field
Interacting nodes are connected with red line. The likelihood of the interacting nodes is penalised proportional to its closeness to the neighbour node.

The defined interaction function which considers just the distance d_{mn} is not sufficient and has a limitation. If a spurious neighbour node n is detected (False Positive) because of a tracking error (which is more likely to occur at weak junctions that show a reduced likelihood value) and enters the Markov Random Field, it may

be classified as an interacting node. This will penalise the likelihood for any correct (True Positive) nodes in close proximity, which is undesirable. To reduce this effect caused by a False Positive node entering the MRF, the interaction function is altered to include the likelihood of the neighbour node such that the interaction function is directly proportional to the neighbour node's likelihood. This exploits the fact that False Positive nodes tend to show a low likelihood value and thus, by including the neighbour's likelihood in the interaction function, evaluate to a low interaction penalty. By doing this, the False Positive node with a low likelihood value in the MRF will be heavily penalised by the strong likelihood neighbours giving it a chance to move out of the MRF and acquire its lost position. The modified interaction function for target *m* with neighbour node *n* at time *t*, $\psi(X_{mt}, X_{nt})$ is:

$$\psi(X_{mt}, X_{nt}) = p_n \times \exp(-f(X_{mt}, X_{nt})) \tag{17}$$

where p_n is the likelihood for node *n*

MCMC-MRF tracker tracking of a set of nodes on a sequence is shown in Figure 24, where the weak junctions (circled in yellow) are not accurately tracked, but are as close as possible to the actual junctions. These are, however, refined during the energy minimisation of the network snakes as explained later in the chapter (section 3.5.4).

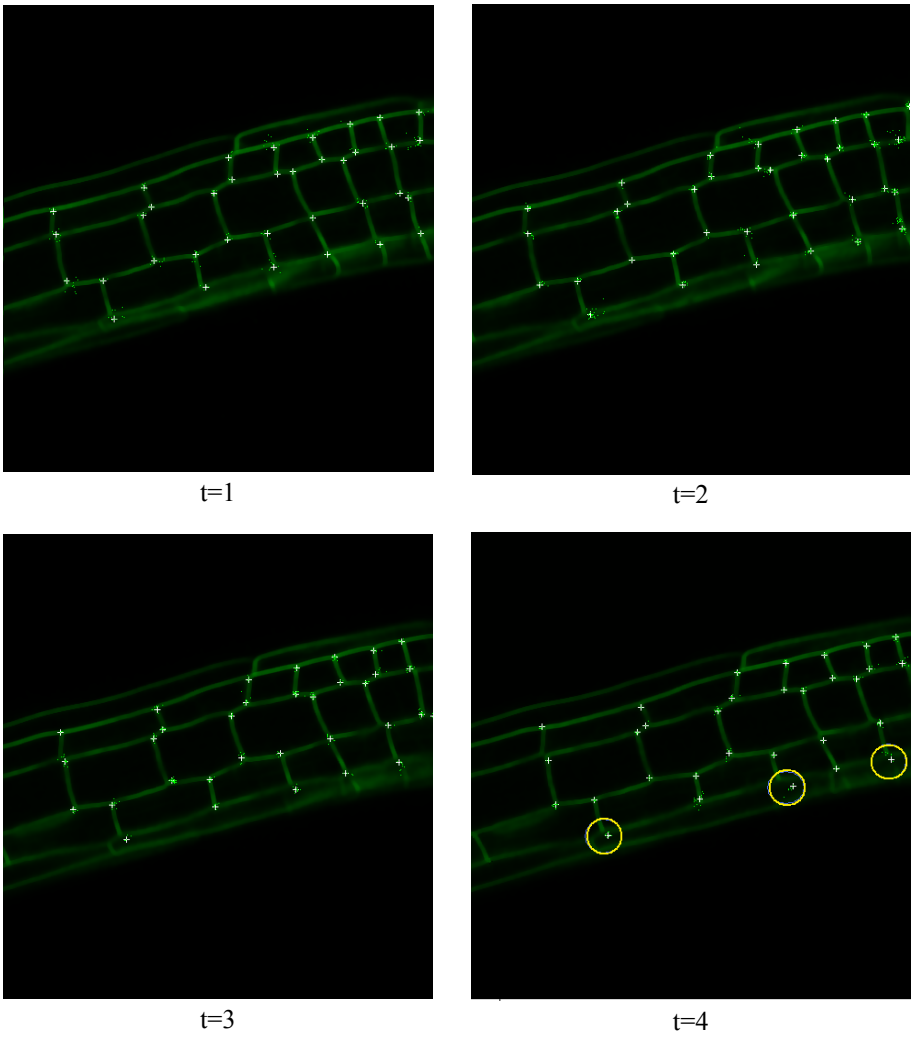


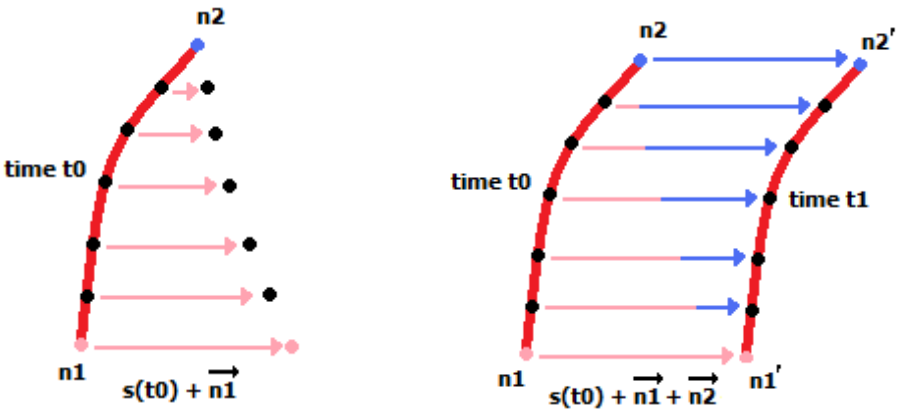
Figure 24. MCMC-MRF Tracker tracking a set of Nodes.
Note that the weak junctions (circled in yellow) are not accurately tracked, but are close to the actual junctions.

3.5.2. Warping the Network

After the network node positions are predicted by the MCMC-MRF based tracker, the previous network snakes topology is then warped to fit the new node positions predicted by the MCMC-MRF tracker.

Consider a snake s(t0) at time t0 with nodes n1 and n2 which is to be warped to new nodes positions n1' and n2' at time t1 as shown in Figure 25. The snake points are warped twice, once for each end node of the snake. Starting from node n1, all the

snake points are warped until the other node of the snake, n2, is reached, such that the nearest snake point to the node of consideration n1 gets the maximum displacement from the displacement vector $\overrightarrow{n1}$ and the farthest snake point to the node n1 gets the least displacement. Similarly, snake points are warped again considering the node n2.



(a) (b)

Figure 25. Warping of Network Snakes
(a) After considering node n1 (b) After considering node n1 & n2

Warping of network snakes is explained in Figure 25. The snake s(t0) at time t0 is displaced to new position at time t1 after applying the warping process. The new positions for nodes n1 and n2 at time t1 are known from the MCMC node tracker. Snake points per snake are warped twice considering one node at a time to fit the snake for the new positions of the nodes. In Figure 25, the displacement caused by each node is distinguished by drawing lines with different colour (pink for displacement caused by node n1 and blue for displacement caused by node n2).

After following the warping process for all the snakes in the network, the whole network is restructured to fit the new network nodes positions obtained from the MCMC node tracker.

The result is effectively a linear interpolation of displacements for the internal snake points, based on distance from the two end junction nodes.

3.5.3. Energy minimization of the Network snakes.

After the network topology is warped to fit on to the new image, the warped network snakes are then allowed to find their own energy minimum. The method used is that described in section 3.4. The energy terms for internal snake points, node points joining three or more cell walls and node points on the image boundary are as in equations (5), (13) and (11) respectively.

3.5.4. Refinement of the MCMC joint state particle distribution

When the network snakes structure has reached its energy minimum, the network nodes may have shifted. The new node positions are the outcome of a global model-matching process, rather than local tracking, and may therefore be expected to improve on the MCMC algorithm's initial independent estimates. The tracker's dynamic model for each target node is updated with the new resultant velocity, and the joint state particles are shifted to allow the new node estimates to be propagated into the next time step - the MCMC tracker is re-initialised with the translated particle distribution and dynamic model.

The NS-MCMC MRF-based hybrid tracker tracking a network of cells on a sequence is shown in Figure 26, where the MCMC tracked nodes at weak junctions are refined and improved (circled in pink at time t=4) compared to the pure MCMC-MRF

tracker output on the same sequence shown in Figure 24 where weak junctions (circled in yellow at time *t*=4) were not tracked accurately.

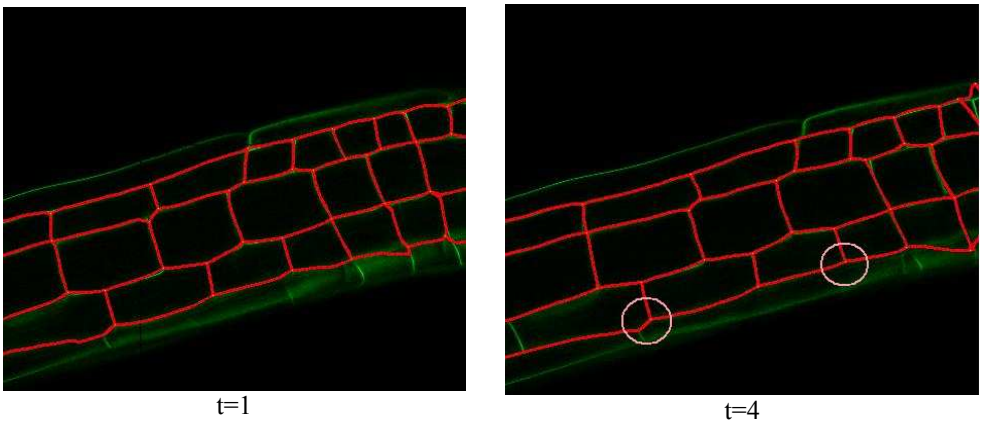


Figure 26. NS-MCMC MRF based hybrid tracker – Output
(Note the output at t=4, MCMC tracked nodes at weak junctions (circled in pink) were refined. Compare with Figure 24)

4 Development of NS-MCMC MRF based Tracker

During the development of the NS-MCMC MRF based hybrid approach presented in the previous chapter, several incremental tracking techniques were developed as a stepwise refinement towards achieving an effective tracking. These different trackers are presented in this chapter, along with a summary discussion on their individual limitations, which led to the development of the NS-MCMC MRF based hybrid tracker described in the previous chapter. This is further extended to cope with varying number of targets as described in chapter 7.

Only a summary discussion on these incremental techniques is presented in this chapter. A more detailed and structured evaluation of the individual component techniques and the resultant hybrid NS-MCMC MRF based approach is presented in the evaluation results chapter, chapter 6.

4.1. Step 1: Development of a pure NS Tracker

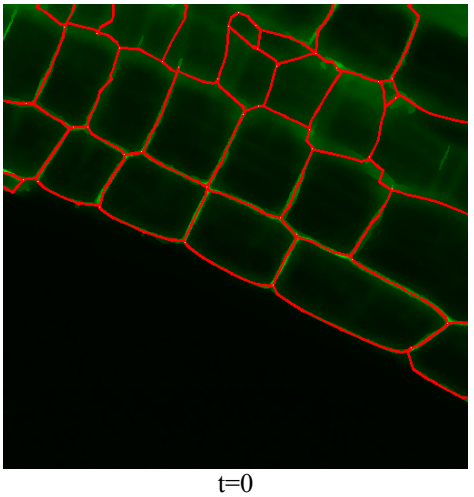
Butenuth's Network Snakes technique is used to model the root structure of Arabidopsis, in which the cells are clustered together and delineated by shared cell boundaries forming a network topology. The topological arrangement of the cells made Network Snakes (NS) a very suitable representation. As with any active contour based technique, the implicit tracking ability of Network Snakes can be used

to create a pure NS tracker capable of tracking networks of cells over time. Figure 27 is an example output of the NS Tracker.

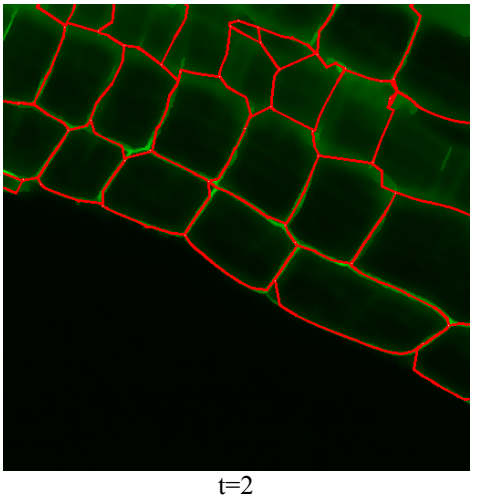
The pure NS Tracker algorithm is as follows

1. Initialise the network snakes data structure using a two-level watershed transform as explained in section 3.3.
2. Allow the network snakes to minimise their energy until the contours reach an energy minimum, so that, the initial configuration of the network is attracted to the features like cell boundaries and nodes to fit the actual image data at time t. The energy equations are described in (5), (11) and (12)
3. Proceed to the next image in the sequence. The energy minimised network in the previous step is used as the initial configuration for the next image in the sequence, time t+1.
4. Go to 2

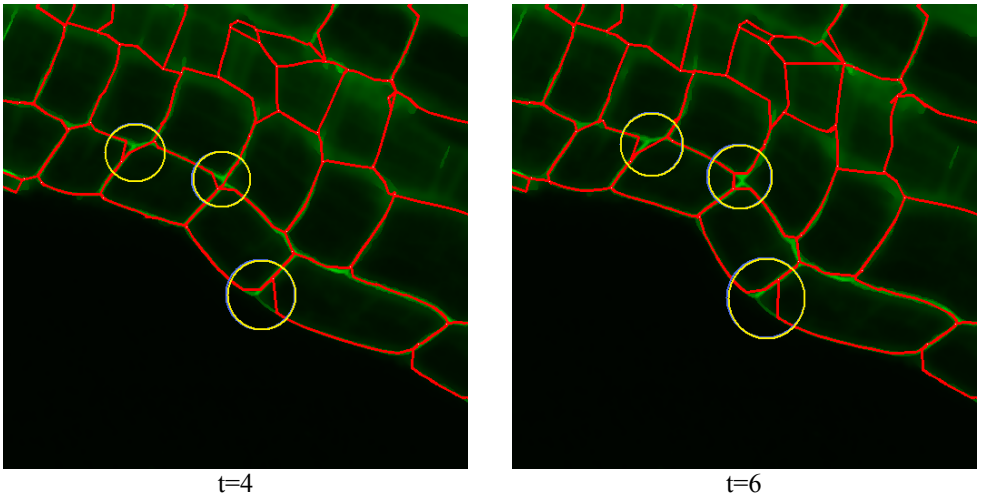
Algorithm 4. NS tracker



t=0



t=2



t=4 t=6

Figure 27. NS Tracker – Output on an artificially warped sequence to simulate extreme bending of root cells (Errors circled in yellow)

4.1.1. Limitations of pure NS Tracker

The tracker performed well in tracking the cell boundaries as seen in the outputs of the NS tracker on an artificially warped sequence created to simulate extreme bending of root cells and given in Figure 27. However, the tracker did not perform well in tracking network *nodes* as highlighted (circled in yellow) in Figure 27, in which the nodes tend to lie on a cell wall with high image gradient rather than settling on a junction point. Image gradient or intensity based external energy is generally a sensible approach, however, as mentioned in earlier chapters, edge detection methods used to estimate the image gradient assume the presence of a step edge in intensity; this is valid along the majority of a snake but, crucially, not at the junction nodes. So, the output of the Sobel and similar edge detection operators or a direct intensity measure is therefore unpredictable at junctions and may not place the nodes correctly as explained in the section 3.4.2. To overcome this problem, an appearance-based external energy is used at junctions. Implementing the modified version of the Network Snakes using an appearance-based external energy is used at junctions led to the next version of the NS Tracker.

4.2. Step 2: Development of NS Tracker with appearance based energy at nodes

By using the appearance based energy at nodes as explained in section 3.4.2, the quality of the NS Tracker in tracking the network nodes has improved as seen in the example tracking output in Figure 28. Here in which the nodes are much more accurately tracked over the sequence (compare with the outputs on the same sequence by the previous version of the tracker shown in Figure 27). The node points are clearly more tightly coupled to the underlying cell wall intersections and, as a result, the sections of snake adjacent to the nodes are a better fit to those walls.

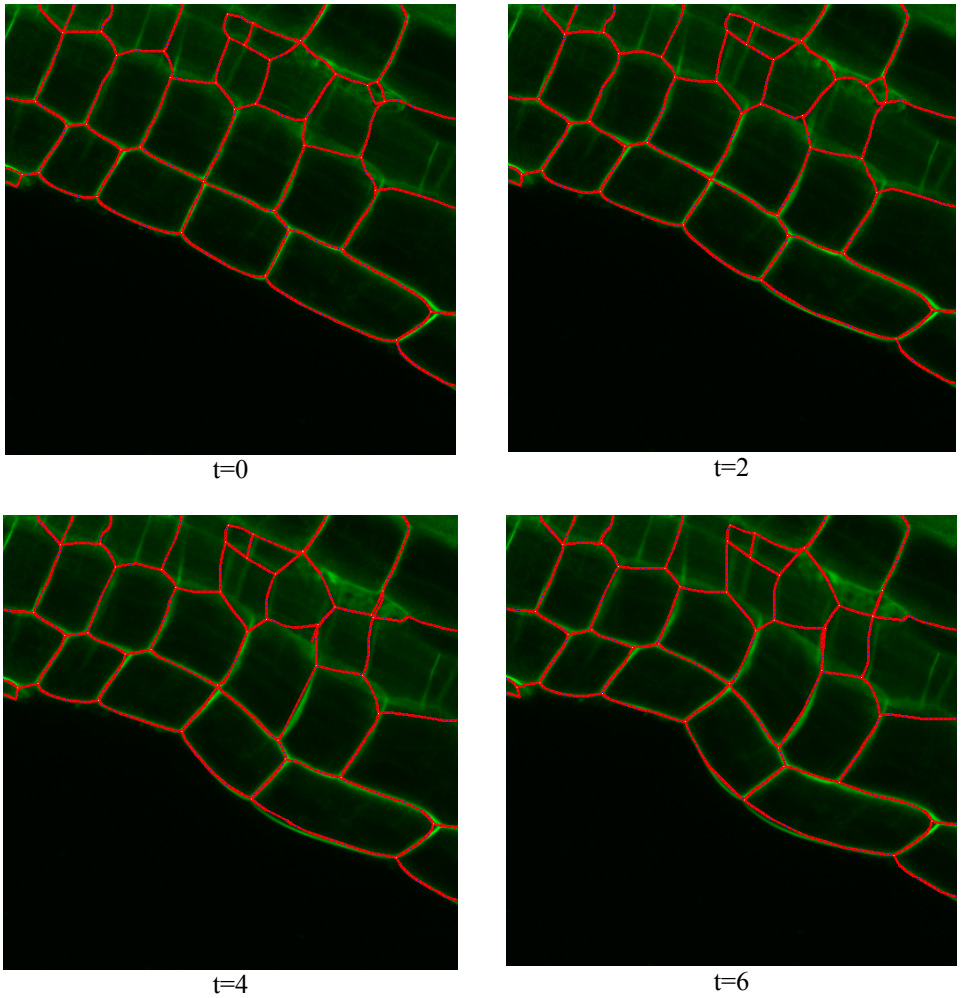
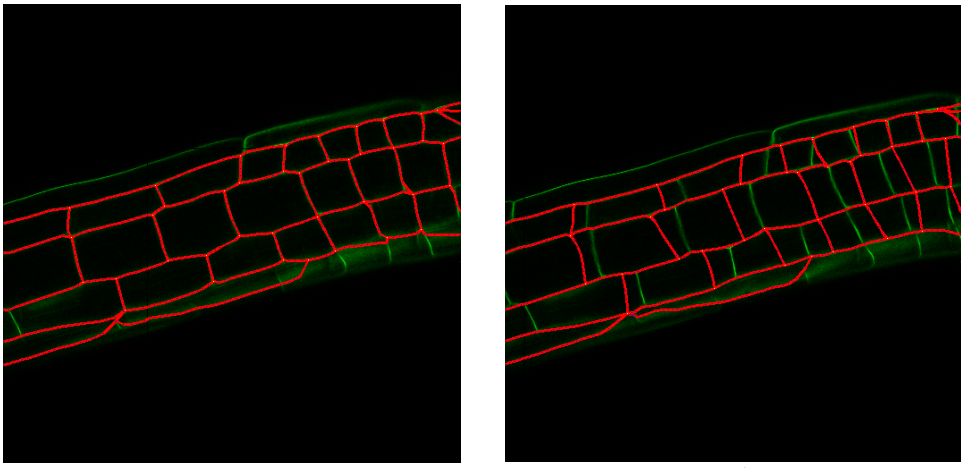


Figure 28. NS Tracker with an appearance based energy at nodes – Output



t=0 t=1

Figure 29. NS Tracker with an appearance based energy at nodes – Failure of the tracker to cope with network shift due to large movements

4.2.1. Limitations of the NS Tracker with appearance based energy at nodes

The tracker performed well in tracking the cell boundaries and the network nodes as long as there was only a small movement of cells between frames as in the example shown in Figure 28; however, the tracker has not performed well for the example shown in Figure 29 which is a time based sequence, where the cells exhibit a large movement between frames. For pure active contour based techniques to work properly, the initial configuration of contours needs to be very close to the actual features [60] [62]. Due to the large movement of cells which causes a significant shift in the whole network, the initial energy-minimised configuration of the network snakes at the previous time step becomes invalid, being located far from the underlying image features and making the implicit tracking eventually fail. Given fast-growing plants like Arabidopsis imaged at high magnification, or when images can only be acquired at quite low frame-rates, as is often the case in CLSM, this situation is common.

To cope with this large movement of cells, a dedicated particle filter based tracker was employed to track the network nodes within the network snakes topology. Network nodes are treated as anchored points on the network and these particle filter-tracked nodes are used to control and modify the network snakes topology for each time step.

4.3. Step 3: Development of a NS tracker using MCMC for tracking nodes

An initial version of the hybrid approach, referred to as the NS-MCMC Tracker was developed using a Markov chain Monte Carlo (MCMC)-based technique for tracking the nodes and using Network Snakes (NS) for tracking *only* the cell walls anchored at the predicted nodes of the MCMC tracker. Note that this initial version did not include the MRF motion model proposed by Khan et al [23].

The MCMC-based particle filter tracker is employed to track the network nodes which are then used to control and modify the network snakes topology at each time step. At each frame, the network snakes are warped to fit the new locations of the network nodes predicted by MCMC tracker. The warped network is then allowed to find its energy minimum; the initial configuration of the network is attracted to the cell boundary features to fit the underlying image data. Nodes are anchored at the predicted positions of the MCMC tracker and only the snake points are allowed to move during this energy minimisation process. Figure 30 shows an example output of this NS-MCMC Tracker.

NS-MCMC Tracker algorithm proceeds as follows

1. Initialise the network snakes data structure using a two-level watershed transform as explained in section 3.3.
2. Allow the network snakes to minimise their energy until the contours reach an energy minimum so that the initial configuration of the network gets attracted to the features like cell boundaries to fit the actual image data at time t. The energy equations are as in (5), (11) and (13).
3. Proceed to the next image in the sequence, time t+1.
4. MCMC tracker is allowed to predict the nodes positions in the new image. The MCMC algorithm is explained in section 3.5.1.
5. Network snake is warped to fit the new locations of the network nodes predicted by the MCMC Tracker. Network warping is explained in detail in section 3.5.2.
6. Minimise the energy of the active contours within the network snake, but not the network nodes; node tracking is purely the responsibility of the MCMC algorithm.
7. Go to 3

Algorithm 5. NS - MCMC tracker

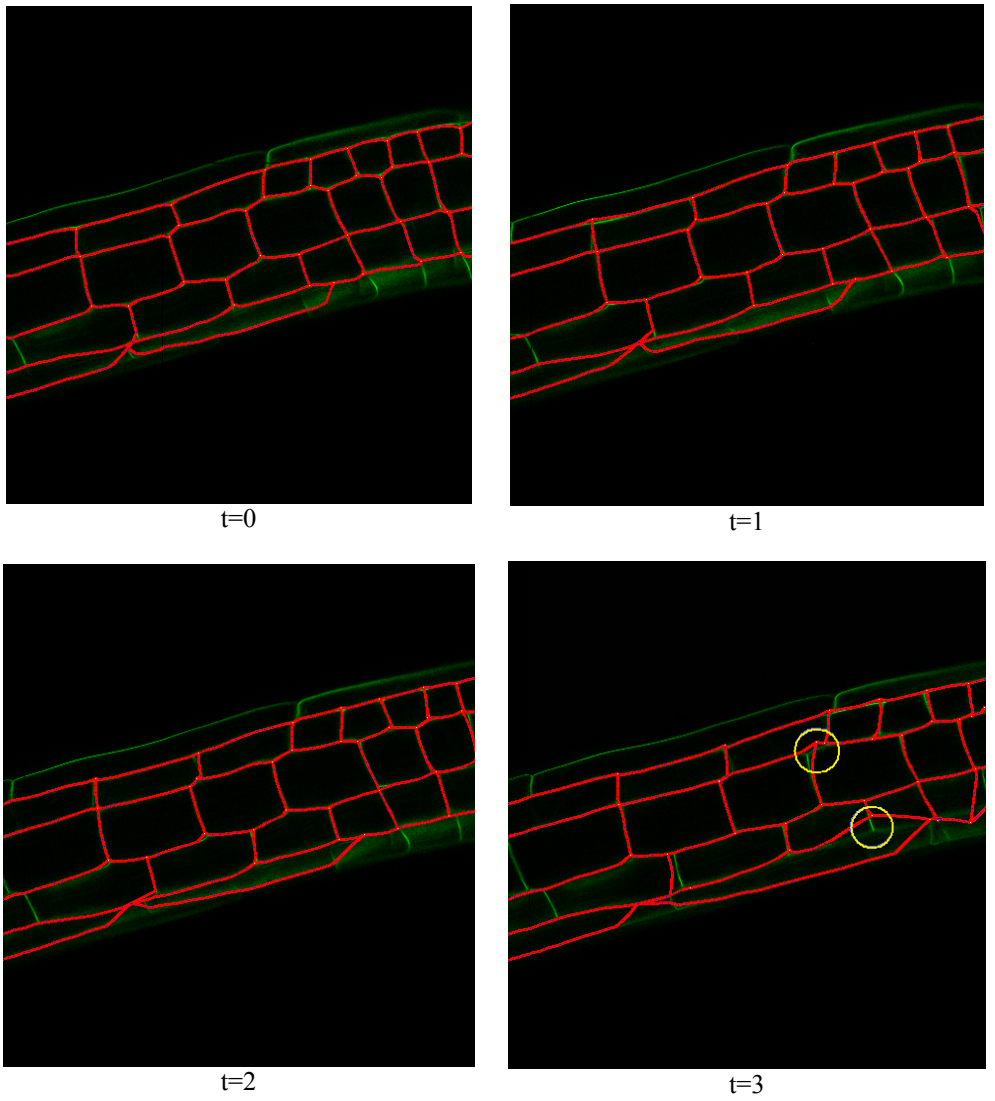


Figure 30. NS-MCMC Tracker on a time based sequence – Output (Node tracking errors circled in yellow)

4.3.1. Limitations of NS-MCMC Tracker

The hybrid tracker is able to control the network topology between frames and track large movements as clearly seen in Figure 30 (compare with outputs in Figure 29). The hybrid tracker performed well in tracking cell walls, but some of the predictions of the nodes are not exactly on the actual nodes as shown (circled in yellow) in Figure 30. This is because if the cell boundaries forming the node are unclear or obscured, the measurement model shows a lower likelihood for all the candidate

points for a given node resulting in an inaccurate prediction. The mean configuration in the MCMC node tracker, however, always keeps the prediction as close as possible to the actual node. For weaker junctions, as shown in Figure 31, the candidate particles tend to cluster around a point that shows some likelihood for a node and the mean configuration is more likely to lie within this cluster of particles making it as close as possible to the actual node. This small error has to be corrected before proceeding to the next frame to avoid the accumulation of the error over the next frames. This can be achieved by allowing the energy minimisation of the network snakes to refine the node positions by making the cell boundaries and also the nodes settle at the energy minimum. This led to the next version of the NS-MCMC tracker with improved tracking at nodes.

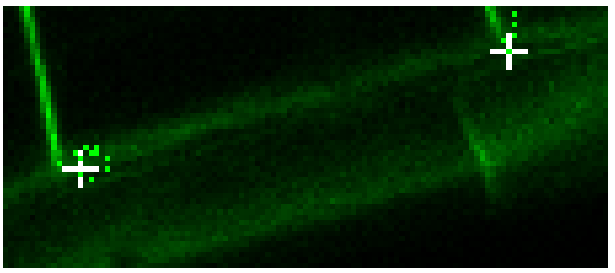


Figure 31. Mean Configuration for MCMC tracker at weak nodes

4.4. Step 4: Development of NS-MCMC Tracker with improved tracking at nodes

Here, the MCMC based particle filter tracker is employed to track the network nodes which are then used to control and modify the network snakes topology by warping the network snakes to fit the new locations of the network nodes as predicted by MCMC tracker. The network snake is then allowed to find its energy minimum and becomes attracted to the cell boundaries. During snake energy minimisation, the *network nodes* are also allowed to move unlike the previous approach in which the

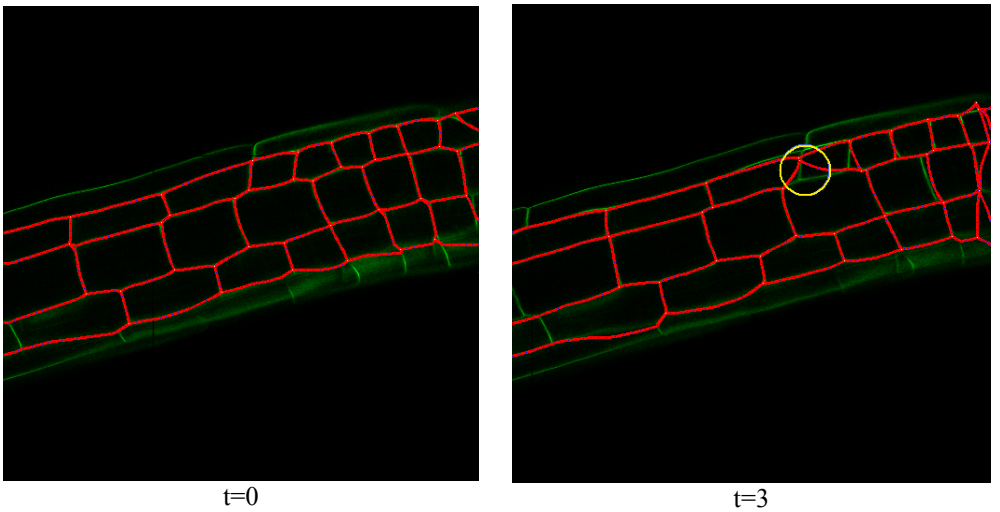
network nodes are anchored during energy minimisation. This refines the MCMC-predicted node positions to fit the underlying actual image data. This corrects any small errors in the MCMC tracker predictions using the global fit of the network snake, and aims to avoid the accumulation of tracking errors over the following frames. The energy function used at the junction nodes is as in equation (13). In this version of the proposed tracker, both the MCMC and the NS technique complement each other. This is in contrast to the previous version of the hybrid tracker, where the MCMC output takes control over the NS. Figure 32 is an example output of NS-MCMC tracker with improved tracking at nodes.

Algorithm for NS-MCMC Tracker with improved tracking at nodes is as follows.

1. Initialise the network snakes data structure using a two-level watershed transform as explained in section 3.3.
2. Allow the network snakes to minimise their energy until the contours reach an energy minimum so that the initial configuration of the network will get attracted to the features like cell boundaries and nodes to fit the actual image data at time t=0. The energy equations are as in (5), (11) and (13).
3. Iterate over the following steps
    - Proceed to the next image in the sequence.
    - MCMC tracker is allowed to predict the nodes positions in the new image. MCMC algorithm is explained in section 3.5.1.
    - Network snake is warped to fit the new locations of the network nodes predicted by the MCMC Tracker. Network warping is explained in detail in section 3.5.2.

- Allow the network snakes to minimise their energy until the contours reach an energy minimum so that the initial configuration of the network gets attracted to the features like cell boundaries and nodes to fit the actual image data at current frame. This will also refine the positions of the MCMC predicted nodes to fit the underlying actual image data.
- Update the dynamic model and the particle set of the MCMC nodes tracker with the new refined positions of the nodes resulting from the energy minimisation of the network snakes.

Algorithm 6. NS-MCMC tracker with improved tracking at nodes



t=0 t=3

Figure 32. NS-MCMC Tracker with improved tracking at nodes – Output
(Merging of two nodes in close proximity, circled in yellow)

4.4.1. Limitations of the NS-MCMC Tracker with improved tracking at nodes

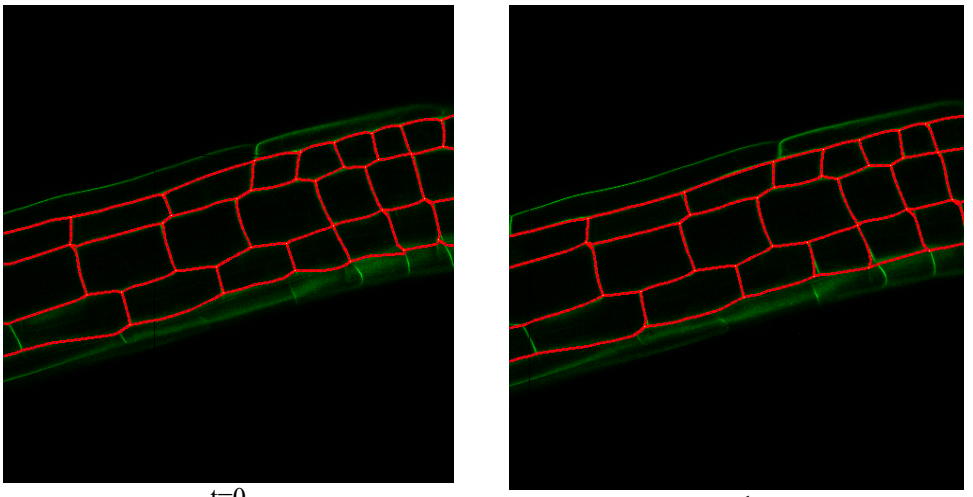
In this version of the hybrid tracker, the component techniques of MCMC and NS complement each other due to which the accuracy of node tracking is improved from the previous version as clearly seen from the example output shown in Figure 32 (compare with outputs in Figure 30). However, if the nodes are in close proximity, there is a chance that a node with high likelihood will hijack the particles of its neighbour node with low likelihood. Both the trackers will then be tracking the same node, causing two network nodes to merge as shown in Figure 32 (circled in yellow). The real-life merging of nodes can never actually occur in the datasets considered in this thesis and hence, needs to be prevented algorithmically. An attempt to solve this problem using an MRF based approach described in section 3.5.1.2 led to the next version of the hybrid tracker using a MCMC based MRF (Markov Random Field) particle filter [23] for interacting nodes.

4.5. Development of NS-MCMC MRF based tracker with interacting nodes

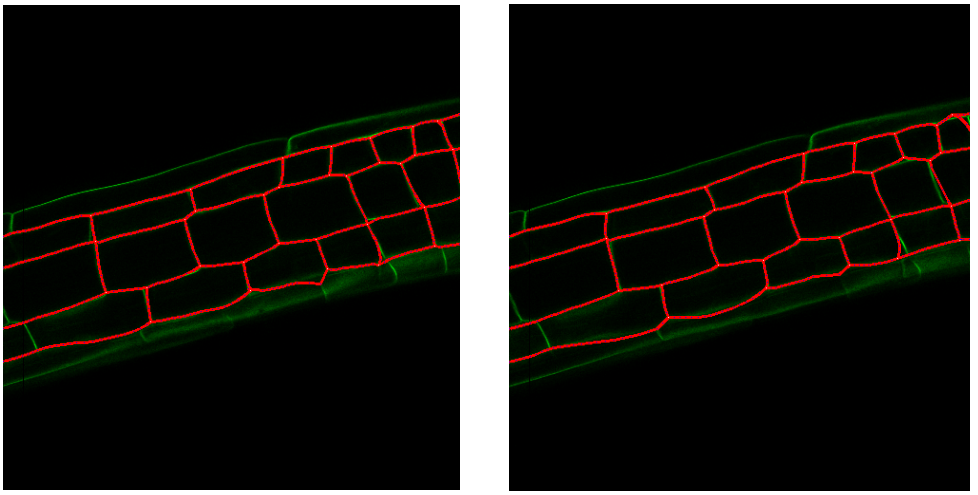
To overcome the problem of merging nodes, a MCMC based MRF (Markov Random Field) particle filter [23] is used to track the network nodes that are now modelled as potentially interacting targets as described in section 3.5.1.2. A Markov Random Field is constructed at each time step, forming a graph of targets that are in close proximity, and the likelihood of particles is penalized by an interaction function proportional to its closeness to another neighbour node. This helps to prevent the tracker from being hijacked by a neighbour node with higher likelihood by keeping the candidate particles away from its neighbour node due to the penalty function. A detailed description of the final NS-MCMC MRF based tracker with interacting nodes is presented in chapter 3.

Some of the outputs of the NS-MCMC MRF based tracker with interacting nodes are provided in Figure 33 (compare with the outputs on the same sequence by the previous version of the tracker in Figure 32) and Figure 34, in which the network of cells are accurately tracked over the time-based sequences.

This version of the hybrid tracker implements the solutions for the problems faced in the approaches discussed so far, such as (i) failure to track large movements, (ii) tracking inaccuracies at network nodes and (iii) merging of targets that are in close proximity. The NS-MCMC MRF based tracker is a robust and accurate solution for the considered real problem of tissue level segmentation and automatic tracking of networks of cells in confocal images. More detailed and structured evaluation of this resulting hybrid technique is presented in chapter 6.

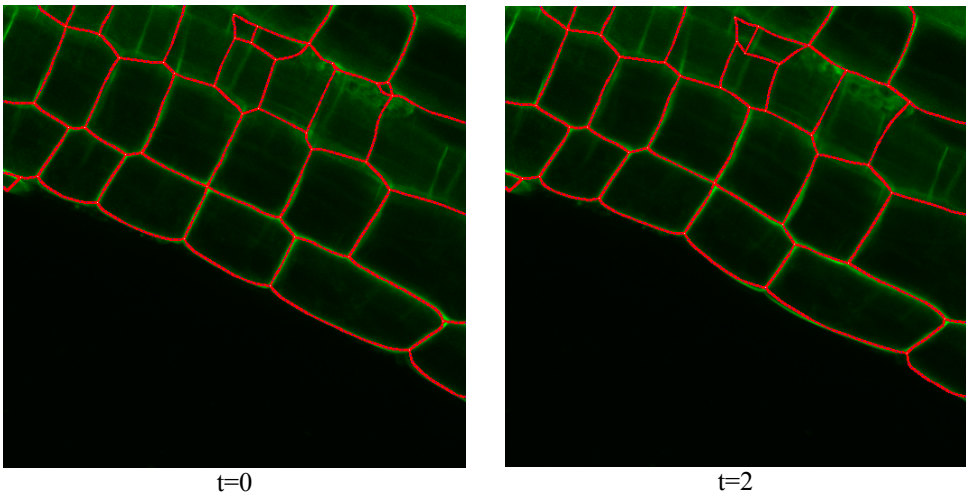


t=0 t=1

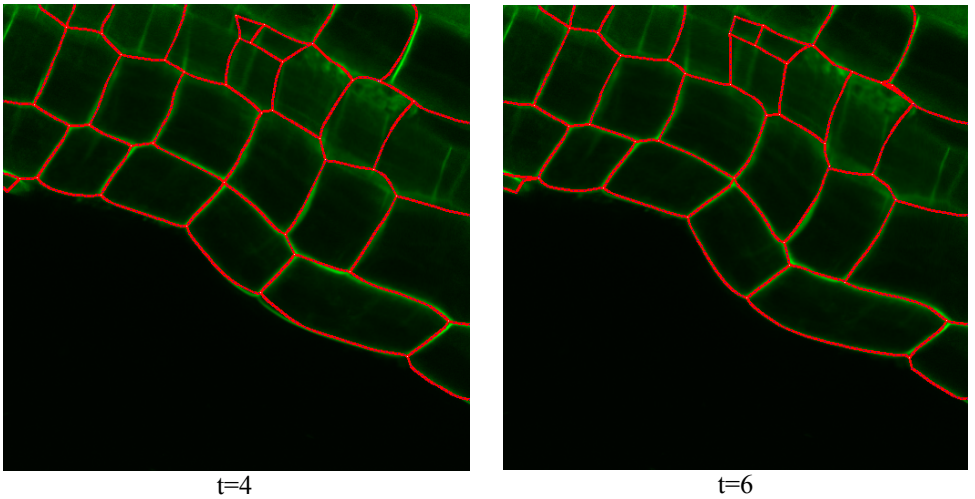


t=2 t=3

Figure 33. NS-MCMC MRF Based Tracker outputs on a time based sequence showing large movements causing a network shift



t=0 t=2



t=4 t=6

Figure 34. NS-MCMC MRF Based Tracker outputs on an artificially warped sequence to simulate extreme bending of root cells

5 Evaluation of Network-Structured Multi-target Tracking Framework

5.1. Overview

To assess and quantify the effectiveness of a tracking technique which is to be applied to real problems, evaluation of the technique becomes absolutely necessary. Evaluation is also useful in comparing different algorithms and techniques. Evaluation of an object tracking technique involves assessing the accuracy and robustness of the tracking algorithm. This chapter will present the evaluation techniques to be used in this thesis.

The evaluation process generally involves comparing the tracker results against some form of ground truth data based on a set of metrics. Choosing appropriate metrics is an important step in achieving an effective performance evaluation. In recent years, more and more research effort has been drawn into performance evaluation of multi-object tracking and the definition of standard metrics for performance evaluation of multi object tracking [113] [114] [115] [116] [117].

PETS (Performance Evaluation of Tracking and Surveillance) workshops are conducted in an effort to provide and analyze standard metrics for performance evaluation of tracking systems [113]. In [113], the authors proposed various object

based metrics and frame based metrics which are then used to evaluate two different trackers and the results are represented using box plots. Correspondence mapping is obtained by drawing a bounding box over the ground truth objects and the system tracked objects and a correspondence is established if the bounding boxes coincide or overlap to a certain degree, which is more suitable for large and independent target objects. In [114], the authors proposed metrics based on True Positive, False Positive and False Negative measures and demonstrated their significance by comparing two different motion tracking algorithms. Metrics and frameworks for performance evaluation of applications such as traffic monitoring and surveillance have also been proposed [115] [117]. In [116], a set of measures and protocols are defined for multi object tracking to measure the accuracy of objects placements and the persistent tracking of an object over time.

Most of the previously defined metrics are based on True Positive, True Negative, False Positive and False Negative measures [113] [114] [115] [116] [117]. Metrics defined for the evaluation of the individual and hybrid tracking techniques in this thesis will also be based on these measures. A brief overview of these measures follows, for completeness.

5.1.1. True Positive

A 'True Positive' or 'True match' occurs when a test which is expected to provide a Positive result does so; that is, when both the tracker results and the ground truth data agree on the presence of an object [113] [118].

5.1.2. True Negative

A 'True Negative' or 'True non-match' occurs when a test is expected to provide a Negative result and does so; that is, when both the tracker and the ground truth data agree on the absence of an object [113] [118].

5.1.3. False Positive

A ‘False Positive’ or ‘False acceptance’ occurs when a test is expected to provide a Negative result, but provides a Positive result; that is, when the tracker suggests the presence of an object, but it is not actually present in the ground truth data [113] [118].

5.1.4. False Negative

A ‘False Negative or ‘False rejection’ occurs when a test is expected to provide a Positive result, but provides a Negative result; that is, when the ground truth agrees on the presence of an object but the tracker does not [113] [118].

5.2. Performance evaluation of component and hybrid tracking techniques

The different types of incremental tracking techniques presented in chapters 3 and 4 will be compared and tested using time-lapse sequences of confocal images of *Arabidopsis thaliana* roots imaged at the Centre for Plant Integrative Biology (CPIB), University of Nottingham. The tracker results are evaluated against semi-automatically generated ground truth data using a set of metrics which will be described in this section.

The evaluation process comprises the following steps.

5.2.1. Ground truth generation

5.2.2. Correspondence mapping between ground truth and tracker results

5.2.3. Computation of evaluation metrics

5.2.4. Presentation of results

5.2.1. Ground truth generation

Normally, ground truth data for object tracking evaluation is generated by manually picking the target objects in a video sequence. Various automatic generation processes have, however, also been used in the past as an alternative to manual ground truth generation. Pseudo synthetic video has been used to generate ground truth data by varying the complexity in the scenes such as adding occlusions [119]. The Open Development for Video Surveillance (ODViS) system, a tool for ground truth construction has also been introduced [120]. ODViS system provides an Application Programming Interface that allows linking the surveillance modules into the framework and a graphical user interface to perform the evaluation using an interactive tool for ground truth definition, visualising the performance of a surveillance module and to compare different modules based on the built-in error metrics.

Fully manual generation of ground truth data is a challenging task, especially for the type of multi-target tracking with which this research is concerned. Manual ground truth generation for such complicated scenes with correspondingly complicated resulting networks is very labour intensive, subjective, and highly prone to fatigue-related errors. Therefore, semi-automatic ground truth generation is employed in the performance evaluation framework in which the semi-automatic initialisation process, which is ordinarily used for initialising the tracker at the first frame as explained in Section 3.3, is used to generate the ground truth data for each frame. The initialisation process is applied to all the frames and the resultant initialised network snake data structure at each frame is followed by a manual correction of any missed or over segmented cells; this forms the ground truth for that frame.

5.2.2. Correspondence mapping between ground truth and tracker results

The aim of the evaluation is to measure the correspondence between the ground truth network and that generated by the tracking system.

For performance evaluation of multi-object tracking, a correspondence mapping between the ground truth data and the multi object tracker results must be established. There are several ways of establishing the correspondence mapping. For example, in [113], correspondence mapping is obtained by drawing a bounding box over the ground truth objects and the system tracked objects and a correspondence is established if the bounding boxes coincide or overlap to a certain degree. However, this approach is more suitable for large and independent target objects, and not suitable for the type of multi-target tracking considered in this thesis where the target objects are connected components forming a network arrangement.

The network data structure under study here is complicated, with many cells connected together via shared boundaries; measuring this correspondence in a meaningful manner is not trivial, and a novel approach is described below.

Simply measuring the displacement errors for nodes does not capture the erroneous deformation in the snake placement, and conversely measuring the quality of the snake alone seems not to place sufficient emphasis on junction location accuracy. On top of this, the concept of the network as a whole fitting the data well needs to be captured. Decomposing the network into multiple snakes and aiming to establish correspondences between the individual snakes of the ground truth data and those making up the tracking output is a feasible solution. This correspondence is established in two stages, first using the node pairs which delimit a snake and then the internal points of which the snake is comprised. Thus, both the junction location and the snake fit quality are measured. Junction nodes and their corresponding snake

are considered together, allowing the higher-level concept of the network, formed out of the snakes and junction nodes, to be incorporated into the evaluation as well.

5.2.2.1. Establishing correspondences between node pairs

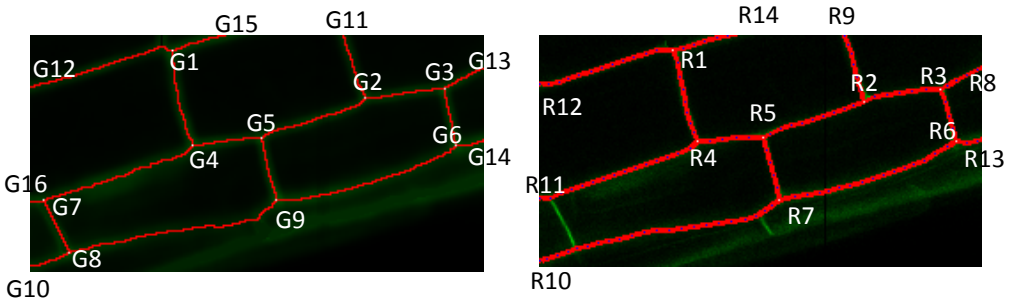
The two open ends of a snake are the nodes, and together form a 'node pair', which is the construct we shall consider first. Correspondences between these node pairs in the ground truth data and the tracker output at each frame are obtained as follows:

1. Each node pair in the ground truth data is mapped to the nearest node pair in the tracker results. If the Euclidean distance separating each matched pair of nodes is less than a fixed threshold (10 pixels in the experiments reported here; average distance error is measured later), a True Positive for node pairs (TP_{np}) is recorded. A visual check is performed to ensure that the matching process has identified the truly corresponding node pair; an error here would be likely to disrupt the network, and so be easy to identify visually.

2. Node pairs in the ground truth data which do not find a corresponding match with the node pairs in the tracker results are marked as False Negatives (FN_{np}) at this stage for that frame.

3. Node pairs in the tracker results which fail to find a match with the node pairs in the ground truth data are marked as False Positives for node pairs (FP_{np}) at this stage for that frame.

Figure 35 below demonstrates a correspondence mapping output for node pairs between the ground truth data and the tracker results, for one frame.



(a) (b)

G1,G4 -> R1,R4	G7,G4 -> R11,R4	G7,G8 -> No Match	No match -> R10,R7
G4,G5 -> R4,R5	G12,G1-> R12,R1	G8,G10 -> No Match	
G5,G2 -> R5,R2	G11,G2 -> R9,R2	G8,G9 -> No Match	
G2,G3 -> R2,R3	G3,G13 -> R3,R8	G16,G7 -> No Match	
G5,G9 -> R5,R7	G6,G14 -> R6,R13		
G3,G6 -> R3,R6	G1,G15 -> R1,R14		
G9,G6 -> R7,R6			

$TP_{np} = 13$	$FN_{np} = 4$	$FP_{np} = 1$
True matches	False rejection	False acceptance

(c)

Figure 35. Correspondence mapping of node-pairs
(a) Ground truth data with all boundaries and nodes (b) Tracker result with G7-G8 node-pair/boundary missing (c) Correspondence map and measures

In Figure 35, there are 16 nodes (G1, G2 ... G16) present in the ground truth data and 14 nodes (R1, R2 ... R14) present in the tracker results: the node-pair or cell boundary G7-G8 is missing in the tracker result (which, in practice, may have been caused by this cell wall appearing in the frame after the network had been initialised, later in the time series). In this example, the match G7,G4 -> R11,R4 is allowed as R11 (on the image boundary) and G7 are sufficiently close such as to be considered correctly placed. This ultimately results in 4 False Negatives and 1 False Positive because of mismatches caused by the nodes connected to the missing nodes in the tracker results.

5.2.2.2. Establishing correspondences between cell boundaries

We next examine the correctly matched node pairs from the previous section to establish correspondences between the proposed cell boundaries of the system results and the actual image data to determine how well the snakes fit onto the actual underlying cell boundary. A cell boundary consists of a set of snake points between two junction nodes. In this stage, these snake points are considered in detail, which is complimentary to the previous stage where only the end nodes were considered.

It is not desirable to directly compare individual ground truth snake points directly with tracked snake points, as the individual points themselves are close together, and yet may be offset by a few pixels from the ground truth, while still producing a good segmentation. Also, due to the elongation and deformation of cells, the snakes change their shape after being initialized at time t=0 and hence, the number of snake points in a tracked snake may be different from that of the ground truth which is from the reinitialised network at each frame. In such circumstances, it is not clear how to choose a meaningful distance measure between the two uneven sets of points to accurately capture the quality of the final snake resting position. Moreover, a distance measure has already been used to obtain the correspondence mapping between junction node pairs in the previous stage which has already provided the nearest matched snakes. It is important at this stage to consider how well the snakes fit onto the actual underlying cell boundary. Therefore, when matching the snake points between the system results and the image data, rather than using the snake point's location, the *image energy* of the individual points is considered, exploiting the fact that the real cell boundaries have high intensity values compared to the inside of the cell. Having approximated the snakes' location using the previous node-pair correspondence, the snake's quality of fit onto the underlying image boundary is checked here.

If any snake point falls off the actual cell boundary in the real image, the image intensity falls off suddenly, and if it falls below a threshold (indicating a sufficient drop in snake quality), a failure is marked in the system track leaving an unmatched snake in the ground truth data and the system track, thereby decrementing the True Positives (TP_{np}) and incrementing the False Negative count (FN_{np}) and the False Positive count (FP_{np}). To calculate a threshold, a local thresholding approach which has proved successful at both weak and strong boundaries as explained in section 3.4.2.1 is used. Using the image energy as a quality measure in this way allows a quantitative evaluation of a snake's fit to the raw image data, given we know the gross position of the snake is correct from the junction node measures in the previous section.

After following this process for all the snakes in the network that gained a correspondence in the previous node pair stage, the resulting True Positives, False Negatives and False Positives are taken as consolidated measures TP, FN and FP respectively as shown in equations (18), (19) and (20). If the total number of mismatches of cell boundaries between the ground truth data and the system track results due to snake points falling off the boundary is denoted by F_s and those suffixed with $_{np}$ are the measures calculated in the previous stage of matching node pairs, then the consolidated measures are:

$$TP = TP_{np} - F_s \tag{18}$$

$$FN = FN_{np} + F_s \tag{19}$$

$$FP = FP_{np} + F_s \tag{20}$$

5.2.3. Computation of Metrics

Further metrics can be computed from the raw TP, FN and FP scores. Metrics are chosen to assess both the sensitivity of the tracker and the associated tracker errors.

For the images considered, True Negatives (the targets which are no longer present) can occur only when the cells leave the image. Though, the tracker allows the cells leaving the field of view to remain connected to the network (as will be explained in section 7.1), they are not considered during evaluation; only the targets within the image bounds are considered for evaluation. So, the tracker does not identify True Negative population, and hence the tracker's specificity cannot be measured.

The set of metrics used in this thesis to evaluate the performance of the tracker are adopted from [113] and are described below.

5.2.3.1. Tracker Detection Rate (TRDR)

The Tracker Detection Rate (TRDR) gives the ratio of identified Positives to actual Positives, also referred to as the *sensitivity* of the tracker. In our case, this represents the proportion of snakes which have their end nodes correctly located and have their snakes following the cell boundary.

The tracker detection rate is expressed as

$$TRDR = TP/TG \tag{21}$$

where,

TP – True Positive

TG – Total number of ground truth objects

5.2.3.2. False Alarm Rate (FAR)

The False Alarm Rate (FAR) measures the Type I error in the tracker, the proportion of false alarms compared to all Positive detections.

The FAR is expressed as

$$FAR = FP / (FP+TP) \tag{22}$$

where,

TP – True Positive

FP – False Positive

5.2.3.3. Object Tracking Error (OTE)

The Object Tracking Error (OTE) is the average distance error in the tracker's estimates of the node-pairs positions. OTE measures the average discrepancy in the location of matched targets (True Positives) in the tracker results to that of ground truth data. Therefore, it must be considered alongside the TRDR, as the OTE is only calculated for successfully tracked targets; it becomes devoid of meaning when the tracker is completely off the target, or no tracker-target association exists.

Object tracking error is expressed as

$$OTE = \frac{\sum_{i=1}^{n} \sqrt{(x_i^{g1} - x_i^{r1})^2 + (y_i^{g1} - y_i^{r1})^2} + \sqrt{(x_i^{g2} - x_i^{r2})^2 + (y_i^{g2} - y_i^{r2})^2}}{TP * 2} \tag{23}$$

where,

TP – True Positives

n – Total number of matched node pairs, in other words True Positives

x_i^{g1} – x-coordinate of the first node in ith node-pair of the ground truth

x_i^{r1} – x-coordinate of the first node in ith node-pair of the system track

y_i^{g1} – y-coordinate of the first node in ith node-pair of the ground truth

y_i^{r1} – y-coordinate of the first node in ith node-pair of the system track

x_i^{g2} – x-coordinate of the second node in ith node-pair of the ground truth

x_i^{r2} – x-coordinate of the second node in ith node-pair of the system track

y_i^{g2} – y-coordinate of the second node in ith node-pair of the ground truth

y_i^{r2} – y-coordinate of the second node in ith node-pair of the system track

5.2.4. Representation of results

Comparing the trackers based on raw metrics such as Tracker Detection Rate, Object Tracking Error, True Positives and False Positives etc. is not very intuitive. Graphical representation of results makes it easier to compare the performance of different trackers. Several ways to represent the results graphically have been used in past research, such as ROC (Receiver Operating Characteristic) curve [117] [118], DET (Detection Error Trade-off) curve [118] and Box plots [113]. The box plot, also known as a box-and-whisker diagram, is used in this thesis for the visual representation of results.

5.2.4.1. Box Plot

This is an intuitive way to visually examine and compare results from different trackers or techniques.

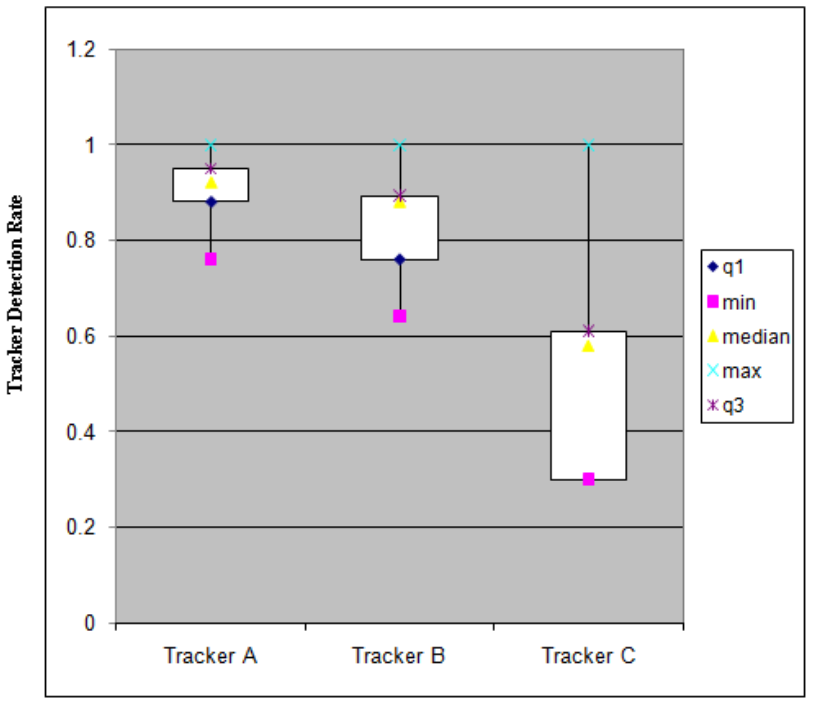


Figure 36. Box Plot

The plot can be drawn for any metric and summarizes 5 significant values obtained from the result set as follows:

1. Minimum value of the result set (min)
2. Maximum value of the result set (max)
3. Median value of the result set (median)
4. Lower quartile (q1) which is the 25th percentile value of the result set
5. Upper quartile (q3) which is the 75th percentile value of the result set

An example box plot shown in Figure 36 is drawn for the metric 'Tracker Detection Rate' for three different object tracking techniques A, B and C applied to the same video sequence. In the example plot, tracker A with higher detection rate, low q1-q3 spread and high median value outperformed the other two trackers.

To summarise, the following three metrics will be used to evaluate the developed tracking framework.

1. Tracker Detection Rate (TRDR)
2. False Alarm Rate (FAR)
3. Object Tracking Error (OTE)

A good tracker should show a high TRDR, a low FAR and a low OTE.

6 Results and Discussion

A NS-MCMC MRF-based hybrid tracker for tracking network-structured, multiple deformable objects has been developed, and the performance of the tracker will now be evaluated on time lapse sequences and 3D z-stack sequences of confocal images of growing *Arabidopsis thaliana* roots, using both real and artificially modified datasets with varying degrees of complexity.

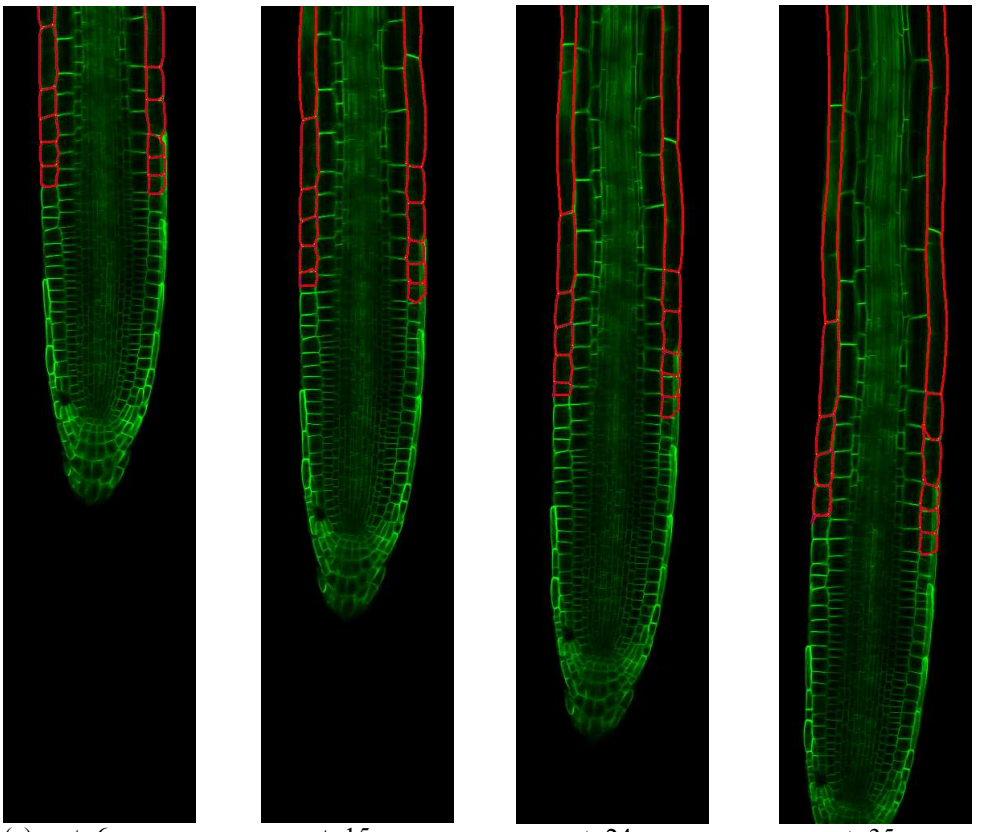
Example frames from the results of running the NS-MCMC MRF-based hybrid tracker on real-life image sequences are shown in Figure 37 and the corresponding evaluation metrics obtained from the tracker are presented in Table 3.

An example frame-wise computation of the metrics TRDR, FAR and OTE for the sequence in Figure 37(a) is presented in Table 4 where, the True Positives (TP), False Positives (FP), False Negatives (FN), Total Tracker tracks (TT) and Total Ground truth tracks (TG) are the numbers of cell wall snakes. Here, varying TT values over the sequence does not mean that the tracker tracks a variable number of targets. In the NS-MCMC MRF based tracking framework, the number of target objects is fixed throughout the sequence and therefore scenarios involving a change in target number, such as might be caused by cells moving into or out of the frame, cannot be meaningfully tracked or any results of such tracking evaluated. To help

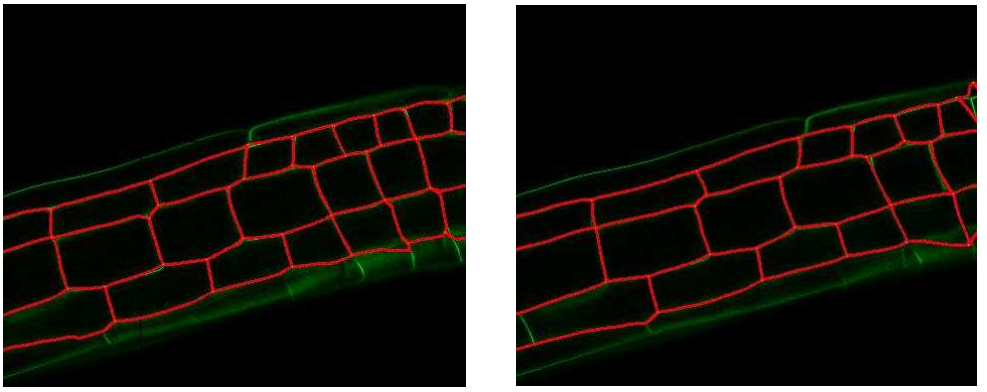
handle this, a *bounding box* is used which prevents analysis of new data entering the image, focusing only on the cells that exist in the image at time *t=0*. Also, trackers leaving the image are no longer considered in the ground truth mapping analysis; they remain connected to the network though, and the results presented included any errors within the bounding box generated by problems caused by this situation, if any. So, at every frame only the targets inside this bounding box are considered for the evaluation and hence the variable number of TT values over the sequence. Expressions for computing these performance metrics are provided in section 5.2.3.

The test datasets were carefully chosen to include sequences of varying complexity, showing large deformation of cells in time based (e.g. Figure 37 (a), (c)) and 3D z-stack (e.g. in Figure 37 (d)) sequences, exhibiting different types of motion behaviours, namely elongation (e.g. in Figure 37 (a), only epidermal elongating cells were chosen for tracking, specifically to highlight the elongation of cells), translation (e.g. Figure 37 (b), (c)) and bending (e.g. Figure 37 (e)).

As shown in the evaluation results in Table 3 and the tracker outputs in Figure 37, the NS-MCMC based hybrid tracker performed consistently well on all the time based and 3D z-stack sequences of CLSM images showing deformation of cells and exhibiting different types of motion modalities. In Figure 37 (a), the network is shown to track elongating epidermal cells over 35 time points. Figure 37 (b-c) demonstrate the ability of the approach to track high magnification images, and hence larger apparent movements between timesteps. The challenge of imaging complete cell wall boundaries without the fading clutter caused by cell walls being just out of the imaging plane is also apparent in these images. The ability of the network snakes to track a bending transformation as well as a translation is evident in Figure 37 (e).



(a) t=6 t=15 t=24 t=35



(b) t=0 t=5

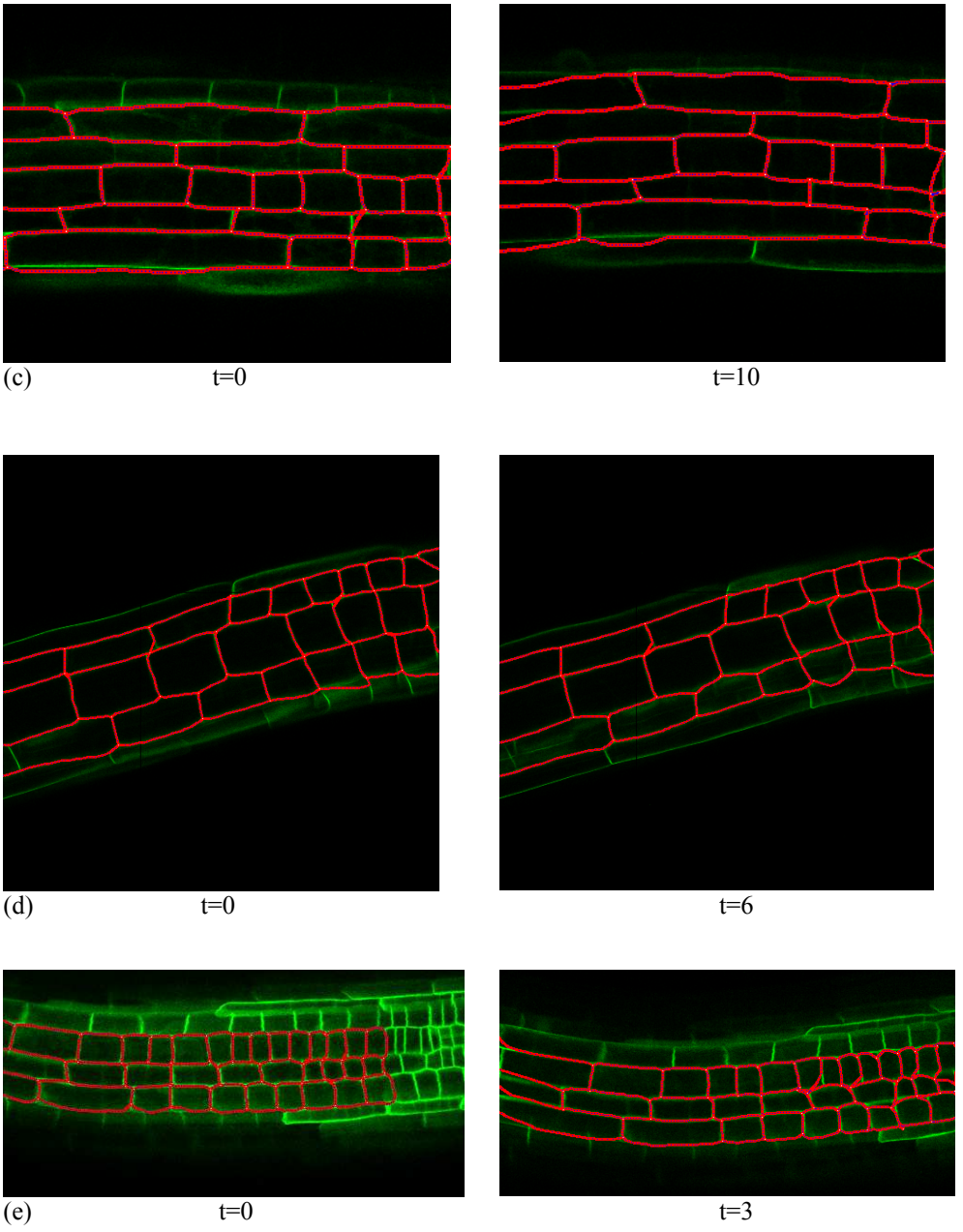


Figure 37. Tracking network of cells showing deformation of cells exhibiting different types of motion behaviour; namely linear, bending and elongating motion.
(a) An elongating root, imaged through approximately the centre of the specimen; (b, c) higher magnification images showing cells elongating/translating across the image plane, imaged through the upper portion of the specimen; (d) images from a 3D z-stack, imaging planes inside the root; (e) an image plane through cells exhibiting a bending behaviour.

Real Image Sequence	TRDR	FAR	OTE
Time based sequence 1 – Tracking cells in the elongation zone Figure 37 (a)	0.94	0. 06	2.34
Time based sequence 2 Figure 37 (b)	0.86	0.15	2.66
Time based sequence 3 Figure 37 (c)	0.85	0.26	2.32
Time based sequence 4	0.85	0.18	4.37
3D Stack sequence 1 Figure 37 (d)	0.88	0.07	2.81
3D Stack sequence 2	0.87	0.11	2.96
Time based sequence 5 –bending motion, Figure 37 (e)	0.89	0.09	2.33

Table 3. Evaluation results on real life image sequences
(Detailed calculation for an example sequence is presented in Table 4)

FRAME #	TP	FP	FN	TG	TT	TRDR	FAR	OTE
1	28	0	0	28	28	1.00	0.00	2.06
2	28	0	0	28	28	1.00	0.00	2.50
3	28	0	0	28	28	1.00	0.00	2.15
4	28	0	0	28	28	1.00	0.00	2.27
5	28	0	0	28	28	1.00	0.00	2.25
6	25	3	3	28	28	0.89	0.10	1.97
7	28	0	0	28	28	1.00	0.00	2.63
8	28	0	0	28	28	1.00	0.00	2.06
9	27	1	1	28	28	0.96	0.03	2.07
10	27	1	1	28	28	0.96	0.03	2.07
11	28	0	0	28	28	1.00	0.00	2.31
12	25	1	1	26	26	0.96	0.04	2.17
13	25	1	1	26	26	0.96	0.04	2.22
14	25	1	0	25	26	1.00	0.04	2.98
15	24	1	1	25	25	0.96	0.04	2.77
16	24	1	1	25	25	0.96	0.04	2.76
17	24	1	1	25	25	0.96	0.04	2.44
18	25	0	0	25	25	1.00	0.00	2.89
19	25	0	0	25	25	1.00	0.00	2.68
20	25	0	0	25	25	1.00	0.00	2.37
21	22	3	3	25	25	0.88	0.12	2.46
22	24	1	2	26	25	0.92	0.04	2.43
23	22	3	3	25	25	0.88	0.12	1.99
24	22	3	3	25	25	0.88	0.12	1.84
25	22	3	3	25	25	0.88	0.12	2.33

26	22	3	3	25	25	0.88	0.12	2.06
27	22	3	3	25	25	0.88	0.12	1.91
28	22	3	3	25	25	0.88	0.12	1.99
29	22	3	3	25	25	0.88	0.12	2.72
30	22	3	3	25	25	0.88	0.12	2.27
31	22	3	3	25	25	0.88	0.12	2.34
32	22	3	3	25	25	0.88	0.12	2.43
33	22	3	3	25	25	0.88	0.12	2.42
34	22	3	3	25	25	0.88	0.12	2.67
	Average					**0.94**	**0.06**	**2.34**

Table 4. Computation of average TRDR, FAR and OTE for the sequence in Figure 37 (a)

These real image sequences were captured via CLSM; the following synthetic image sequences are generated from the real image sequences by adding varying degrees of complexity and deformity in the following manner:

1. Varying the amount of movement between frames
2. Varying the motion behaviour of cells
3. Adding image noise to the data

The purpose of these synthetic sequences is to test the algorithms beyond the situations seen in the real data sets currently captured at CPIB. Given different biological goals and/or imaging protocols, it is likely that when the techniques and associated software are released to the biological community the proposed method will be applied to a wider range of image sequences than those immediately available.

6.1. Varying the movement of cells between frames

In this case, test sequences are generated by temporally subsampling a real life growth sequence, thereby increasing the apparent movement of cells between frames. Here, they are modified in the following ways:

1. Generating a sequence only considering every second frame from the real image sequence

2. Generating a sequence only considering every third frame from the real image sequence

3. Generating a sequence only considering every fifth frame from the real image sequence

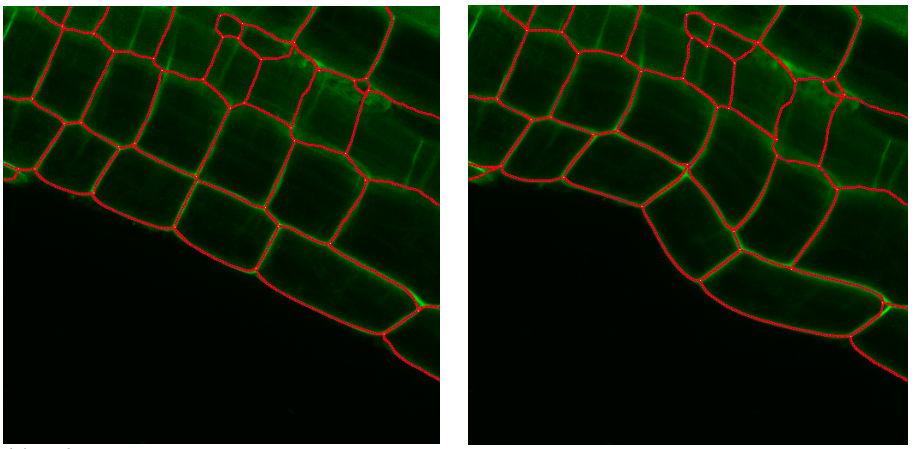
Thus, the sequences become shorter, but the time steps between frames are effectively lengthened, producing the illusion of the root growing much faster than it does in the corresponding real sequence. The evaluation metrics obtained from the original real sequence and the subsampled sequences are tabulated in Table 5. This shows a consistently high TRDR and low FAR and OTE. This clearly shows that the tracker consistently performs well, even over the sequences showing large movement of cells between frames. Interestingly, the TRDR for the sequence using every third frame is *slightly* better than the sequence using every other frame. This is likely to be because the sequence is shortened due to dropping frames; factors such as uniformity in the velocity profile over the generated sequence and the arrangement of the network topology such as closeness of the network nodes in the considered frames might have influenced the results. But, clearly once a movement threshold is exceeded in the final sequence, as expected tracking gets markedly worse. Table 5 demonstrates that the algorithm should be capable of tracking sequences captured within the bounds of practical biological experiments and that the algorithm performed well using at least one third of the sampling frequency of the original sequence.

Test Data	TRDR	FAR	OTE
Real Sequence shown in Figure 37 (a), using every frame	0.94	0. 06	2.34
Sequence generated with every other frame	0.91	0.09	3.24
Sequence generated with every third frame	0.94	0.06	3.05
Sequence generated with every fifth frame	0.72	0.28	3.45

Table 5. Evaluation results on a modified real sequence varying the movement of cells

6.2. Varying the motion behaviour of cells

In this case, the cells in a real image sequence are artificially warped to simulate extreme bending of root cells rather than a translation, and consequently a severe deformation of the network topology, which is similar to those seen in plants whose cell wall development has been disrupted. This disruption can occur due to genetic mutation or changes in hormone concentration and location. It may arise when the cells clog on the surface of the growth medium obstructing the normal growth of the root and causing the root to bend. In contrast to the real bending sequence shown in Figure 37 (e), the motion caused due to this disruption is more complex as the bending occurs only at a particular region of the growing root. As no real sequences showing these effects at high resolution are currently available within CPIB, extreme bending of cells has been simulated using a non-linear warping tool in Adobe Photoshop. The results obtained from such artificially warped sequence are shown in Figure 38 and the numerical results presented in Table 6. The results show that the network structure is able to cope with such complex motion and deformation.



(a) t=0 (b) t=8

Figure 38. Tracking root cells that exhibit a bending motion

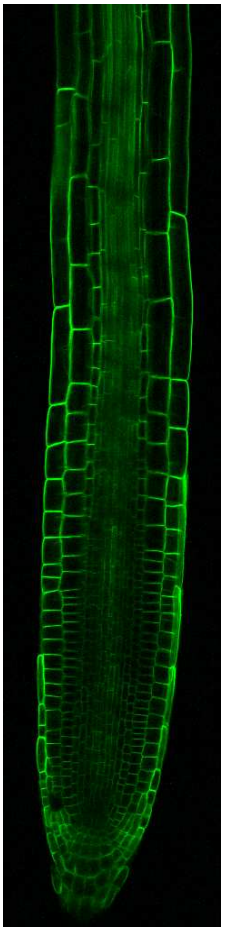
Test Data	TRDR	FAR	OTE
Artificially warped sequence	0.81	0.29	2.17

Table 6. Evaluation results on a real sequence varying the motion behaviour of cells

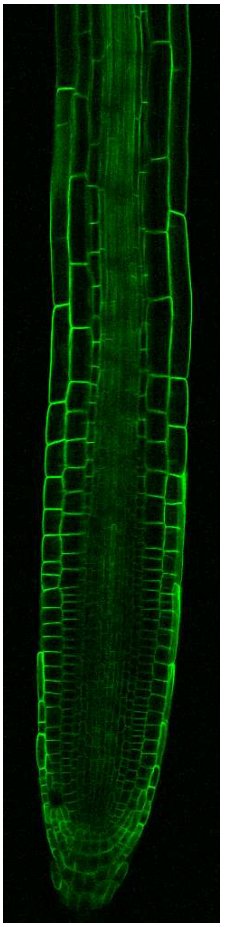
6.3. Adding pixel Noise

CLSM images of Arabidopsis contain impulsive pixel noise which is non-Gaussian in nature as demonstrated in [20] using a Q-Q plot drawn against the theoretical quantiles and the sample quantiles. Three noise components were identified; approximately Gaussian sensor noise, approximately Gaussian biological noise and large outliers. To test the proposed methodology in both realistic and extreme noise conditions, various levels of artificial noise were added to the sequence in Figure 37 (a). A noise distribution Q-Q plot for added synthetic noise level 2 is shown in Figure 39 (i). Noise was added using a realistic distribution [20], comprising two sets of zero-mean Gaussian noise, generally representing detector noise (all pixels, smaller σ) and biological noise (proportion of pixels, larger σ). Additionally, infrequent extreme outliers were added.

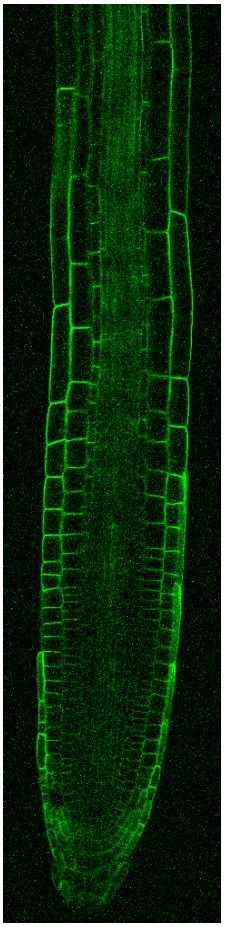
Outputs of the tracker on image sequence with varying levels of synthetic pixel noise are shown in Figure 39 and the corresponding evaluation metrics obtained from the tracker are presented in Table 7. The tracker performed well at all realistic noise levels, and performed capably even under severe noise beyond realistic levels.



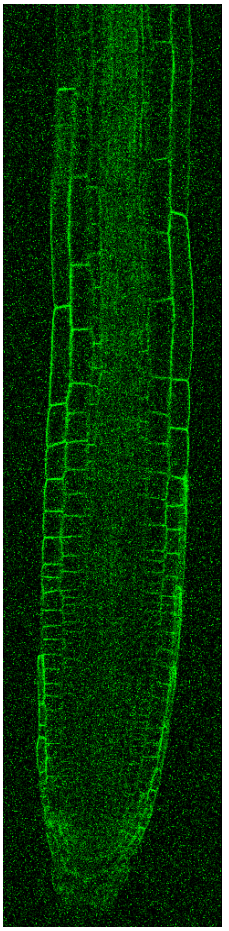
(a) Noise Level 1. Sensor σ = 5, Biological σ = 10, on 10% pixels, Outliers on 0.01% pixels



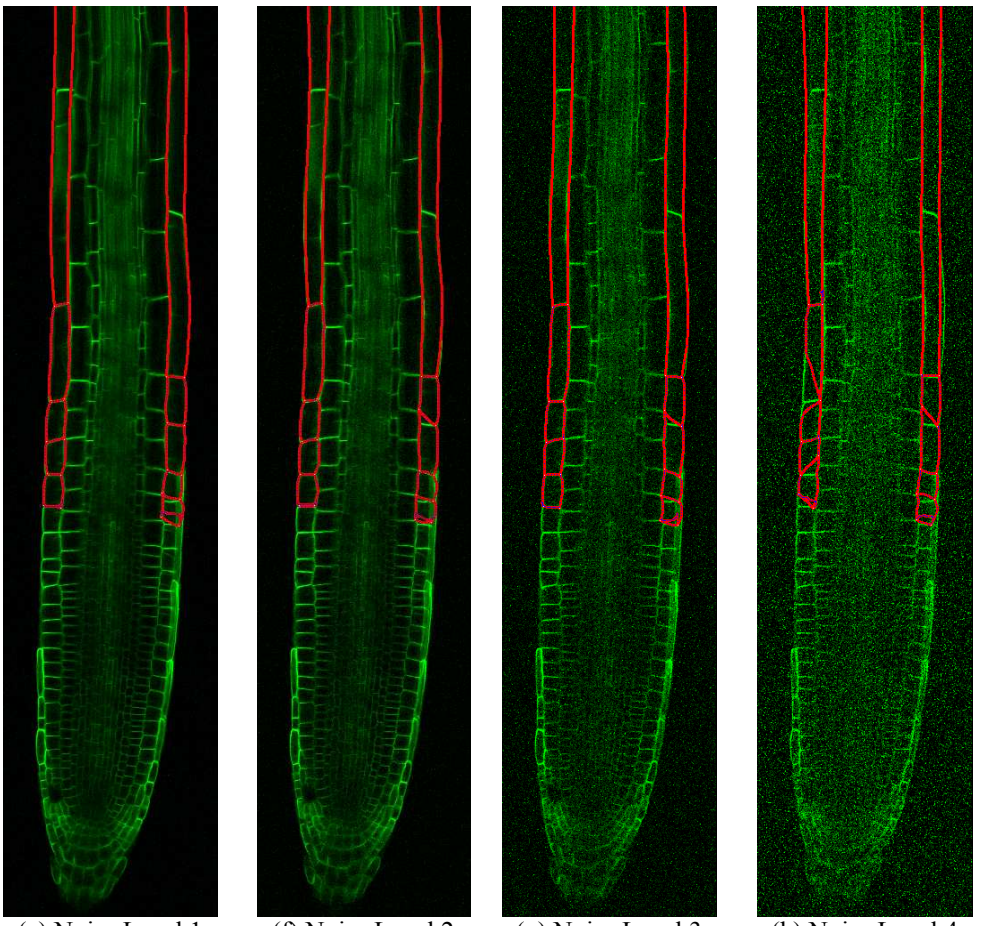
(b) Noise Level 2 Sensor σ = 10, Biological σ = 20, on 10% pixels, Outliers on 0.1% pixels



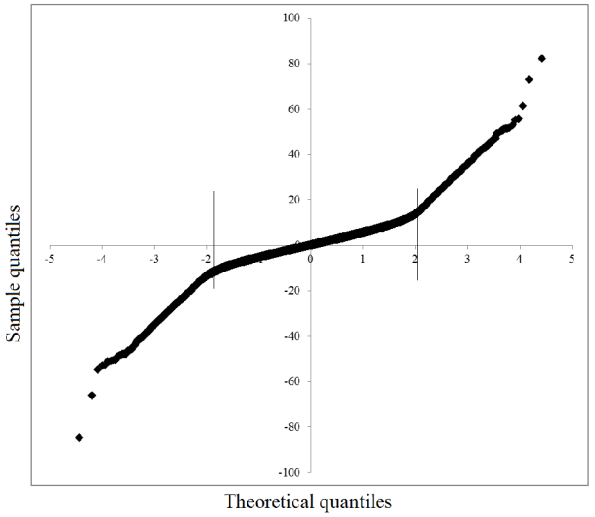
(c) Noise Level 3 Sensor σ = 30, Biological σ = 80, on 10% pixels, Outliers on 1% pixels



(d) Noise Level 4 Sensor σ = 50, Biological σ = 150, on 20% pixels, Outliers on 3 % pixels



(e) Noise Level 1 (f) Noise Level 2 (g) Noise Level 3 (h) Noise Level 4



(i) Example QQ plot of noise distribution for noise level 2

Figure 39. Tracking of root cells in the elongation zone with synthetic pixel noise added. All images at time t=30. (a-d) Input images with increasing levels of noise added; (e-h) Corresponding tracker results (i) Example QQ plot of noise distribution for added synthetic noise level 2 showing approximately Gaussian detector noise (smaller **σ** - central linear plot between marked vertical lines) and approximately Gaussian biological noise (larger **σ**) with infrequent outliers

	NS-MCMC MRF			MCMC		
	TRDR	**FAR**	**OTE**	**TRDR**	**FAR**	**OTE**
Real Sequence With Noise Level 1	0.91	0.09	3.16	0.76	0.24	3.67
Real Sequence With Noise Level 2	0.87	0.12	3.09	0.77	0.22	4.03
Real Sequence With Noise Level 3	0.86	0.14	3.02	0.77	0.23	3.76
Real Sequence With Noise Level 4	0.66	0.34	3.81	0.55	0.43	3.58

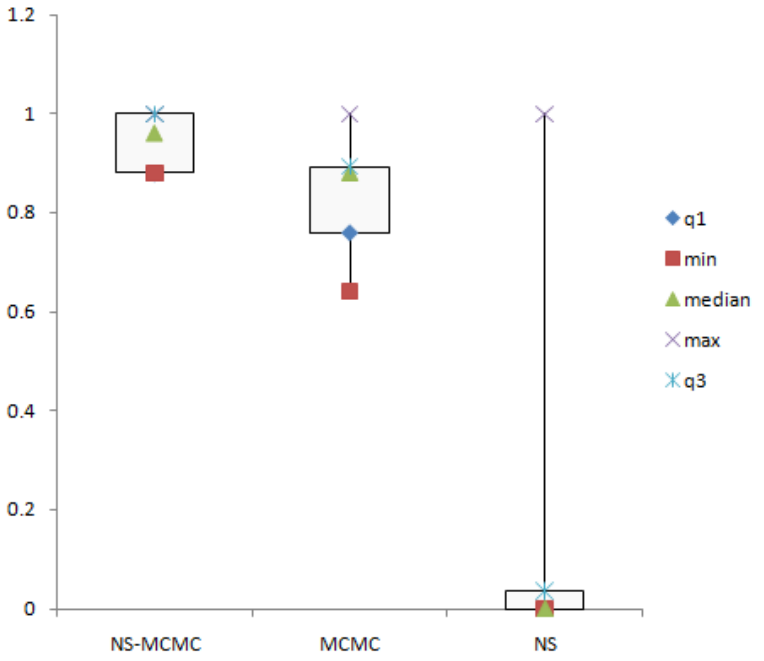
Table 7. Evaluation results on a real sequence with increasing noise levels

6.4. Comparison of the proposed NS-MCMC MRF based hybrid tracker with independent NS and MCMC trackers

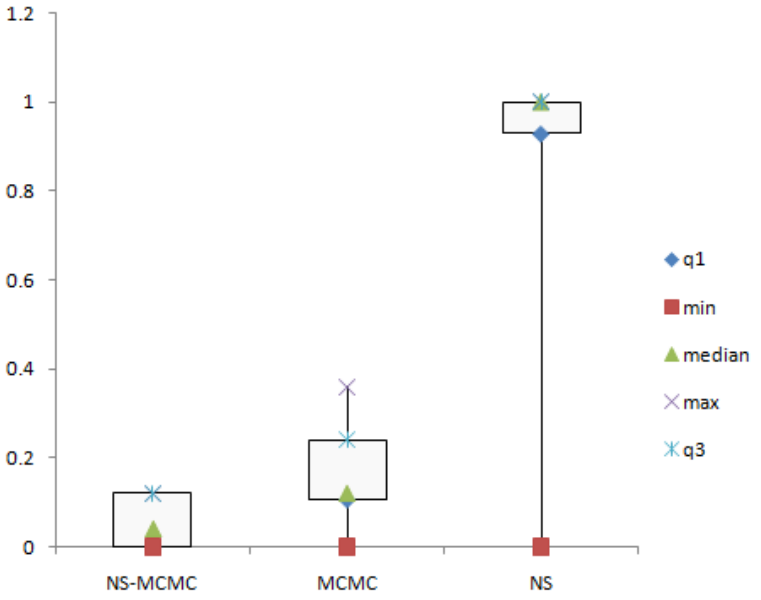
To highlight the performance of the hybrid tracker proposed here, the NS-MCMC MRF based hybrid tracker framework is compared with its component concepts: the network snake's implicit tracking ability alone (section 4.1), and an MCMC tracker tracking junctions (section 3.5.1) independently. All the trackers were run on the image sequence shown in Figure 37 (a) and on the generated sequences with increased movement of cells between frames as explained in section 6.2. The results are tabulated in Table 8 and a graphical comparison of results for a real image sequence (Figure 37 (a)) is presented as box plots in Figure 40. These box plots are drawn for the metrics TRDR, FAR and OTE for all the 3 different tracker variants. The figure clearly shows that NS-MCMC MRF based hybrid tracker with high TRDR values and low FAR and OTE values has performed consistently well, and better than the other two independent trackers. The TRDR box plot shows that the hybrid tracker's TRDR is higher than the other two trackers and is consistent throughout the sequence. The MCMC tracker performed better than the network snakes' implicit tracking alone, whose TRDR is very low compared to the other

trackers. In fact, using network snakes alone resulted in immediate failure on the dataset (for an example failure of the Network Snakes tracker to cope with network shift caused by large movements, see Figure 29) which is why the NS tracker has not been considered for the comparison of the metric OTE in Figure 40(c).

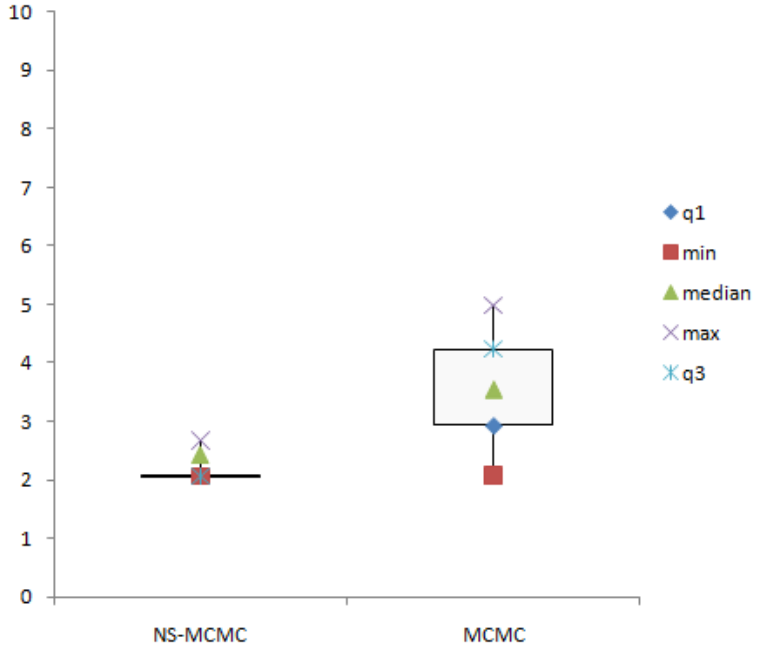
The plot for NS-MCMC hybrid tracker shows a very low OTE values compared to the independent MCMC tracker showing that the hybrid tracker is more accurate than the independent MCMC tracker.



(a) TRDR



(b) FAR



(c) OTE

Figure 40. Box plot comparison of trackers

	NS-MCMC MRF			MCMC		
	TRDR	**FAR**	**OTE**	**TRDR**	**FAR**	**OTE**
Real Sequence	0.94	0.06	2.34	0.84	0.15	3.54
Every 2nd frame	0.91	0.09	3.24	0.81	0.19	3.64
Every 3rd frame	0.94	0.06	3.05	0.84	0.16	3.87
Every 5th frame	0.72	0.28	3.45	0.65	0.36	4.23

Table 8. Evaluation results Comparison of NS-MCMC MRF and MCMC alone tracker

The independent network snakes tracker resulted in a complete failure on the dataset as the network snakes technique itself is not capable of tracking the large movements present in this dataset. As the NS tracker resulted in a complete failure on the datasets, the evaluation metrics obtained from the NS tracker are not tabulated in Table 8, which shows the metrics obtained from the NS-MCMC MRF and MCMC trackers.

The trackers were also run on the real image sequence shown in Figure 37 (e) where the cells deform and exhibit a bending and translation motion, as a bend forms in the growing root. The comparison of evaluation metrics of the combined NS-MCMC MRF based hybrid tracker and the independent MCMC tracker for this sequence is presented in Table 9. This shows that the hybrid tracker again performed slightly better than the independent MCMC tracker. Moreover, the NS-MCMC MRF hybrid tracker tracks the whole network of cells whereas MCMC tracking alone only tracked the network nodes. So, the complete deformation of the cells is tracked by the NS-MCMC MRF hybrid tracker and provides all the information about the network topology such as boundary points of cells and nodes, unlike the MCMC tracker which only provides the locations of the network junction nodes. This is a

major advantage of the proposed hybrid tracker, which has the robust tracking of the MCMC filter, and the fine topological resolution of the snakes.

	NS-MCMC MRF			MCMC		
	TRDR	**FAR**	**OTE**	**TRDR**	**FAR**	**OTE**
Bending sequence - Figure 37 (e)	0.89	0.09	2.33	0.83	0.16	3.36

Table 9. Evaluation results for bending sequence- Comparison of NS-MCMC MRF and MCMC alone tracker

6.5. Limitation of NS-MCMC MRF based hybrid tracker

In the image sequences considered in this thesis, when cells exhibit normal growth, new cells enter at one end of the visible root segment while others leave through the image boundary at the other end of the segment. This results in topological changes and a varying the number of target cells in view. This can be clearly seen in the real image sequences presented in Figure 37 (a), (b), (c) and (e). The NS-MCMC MRF based hybrid tracker which is designed to track a *fixed* number of targets over time, performed consistently well on all the time based and 3D z-stack sequences of CLSM images showing deformation of cells and exhibiting different types of motion modalities; namely linear, elongation and bending motions and with varying levels of realistic and severe, unrealistic noise. It is not, however, capable of handling topological changes between frames caused by cells entering and leaving the image. In what follows, a solution to this problem of tracking *variable* number of targets is presented; in our case, tracking a variable number of cells which causes such a change in the network topology.

7 Tracking Network-Structured, Deformable Objects in Varying Numbers - Entering and Leaving Cells

In confocal images of live *Arabidopsis thaliana*, when roots exhibit normal growth, new cells enter at one side of the image while other cells leave through the opposing image boundary at the other end of the segment as seen in the example shown in Figure 41. Here, only the region inside the bounding box remains the same until the last frame of this four frame sequence and the region outside this bounding box either leaves the image boundary or has entered the image boundary after time t=0. In this example, the cells leave the image from the right side of the image boundary and enter through the left side of the image boundary, as the root is growing left to right. This is a common property of the images taken within CPIB of growing roots.

As new cells enter and some existing cells leave, the number of nodes and cell boundary snakes within the field of view is variable. This demands re-parameterization of the network topology, and the tracking framework needs to be adapted to cater for a varying number of targets to handle such topological changes.

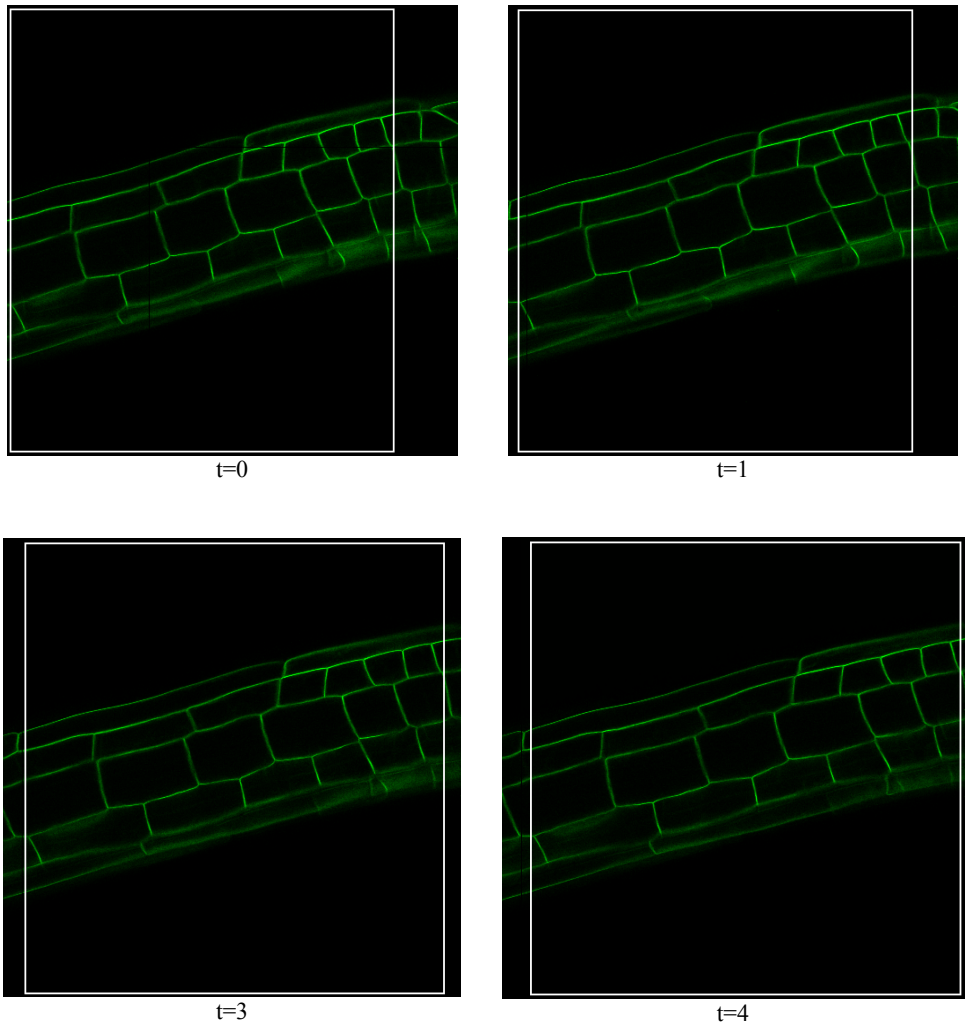


Figure 41. Entering and leaving cells over a sequence
(only the cells inside the white bounding box remain same till the last frame)

The MCMC based filter used for tracking nodes (section 3.5.1) is limited to a fixed dimensional state space which is only suitable for tracking a fixed number of targets. Previous research addressing the issue of tracking a variable number of targets suggests a number of potential solutions to this problem. In [121], the particle filter implementation is extended according to finite set statistics (FISST) and applied to tracking a random sets of vehicles in different types of terrain such as road, field and forest. A particle filter implementation of a Probability Hypothesis Density (PHD) filter has been used in tracking random sets of targets [122] [123]. Although these techniques form potential solutions to this problem, a more convenient attempt to

solve the problem is to extend the MCMC sampling to variable dimensional state spaces by using the Reversible Jump MCMC (RJMCMC) technique [24]. Khan et al [24] used the RJMCMC filter to track a variable number of interacting targets. The method was found to be successful when applied to real image sequences from the social insect domain. In [101], the RJMCMC technique has been used for multi-person tracking that infers the number of persons in the scene, their body and head locations and head pose.

The problem of cells leaving the image boundary is handled first, followed by cells entering the image boundary.

7.1. Cells leaving the field of view

In confocal images of growing Arabidopsis roots, the cells have been modelled as connected components delineated by shared boundaries forming a network topology. In what has been considered so far, this topology has not been allowed to change. If cells that leave the image are removed, their previously-connected neighbour cells will consequently lose one or more boundaries and associated nodes. For example, consider Figure 42 in which the white line marks the original image boundary and the cell walls to the right of this image boundary have left the image; consider removing the cell walls *1* and *2* that are outside the image boundary. Note the cell wall *2* has not completely left the image and some part of the cell wall is inside the image boundary and is connected to the node N1. If the cell wall *2* is removed, the node N1 loses one of its connected cell walls and becomes just a point connecting two cell walls *3* and *4,* which is theoretically no longer considered a node, as a node should either be connected to the image boundary or a junction connecting three or more cell walls. If N1 is no longer a node point, the cell walls *3* and *4* are now connected by a non-node point N1 which violates the theory of network snakes and therefore disturbs the topology.

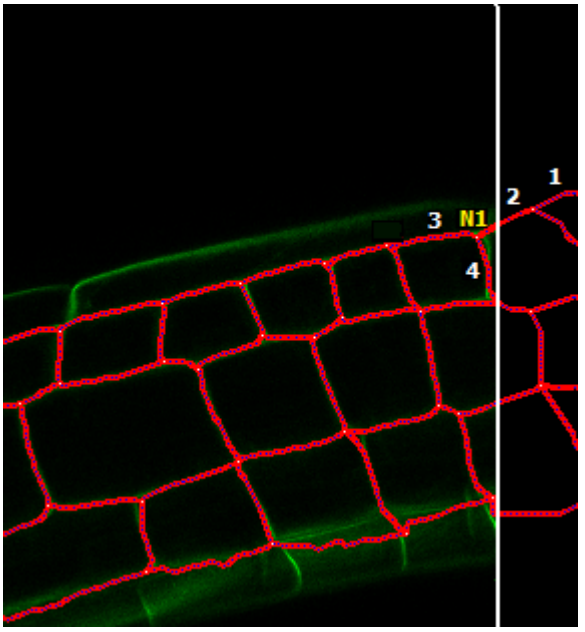


Figure 42. Cells leaving the image boundary
(snake *1* has already left the image boundary while snake *2* has partially left the image boundary. Node *N1* is a common nodal point for snake *2*, *3* and *4*. Losing snake *2* means losing node *N1* as well)

Removing any information from the network might destroy the topology which could cause the network structure to collapse. To resolve this, the cells that leave the image boundary are allowed to remain connected to the network to preserve the topology.

This is achieved in the following steps.

At each frame,

1. The image boundary is extended by a value which is the average velocity of all the nodes in the network along the direction of motion of cells. For example, if the direction of motion is positive and aligned with the *X* axis, then the right side image boundary is extended. For the image sequences considered in this thesis, the direction of motion is either along the horizontal, *X* axis (with a negligible motion along the *Y* axis) or the vertical,

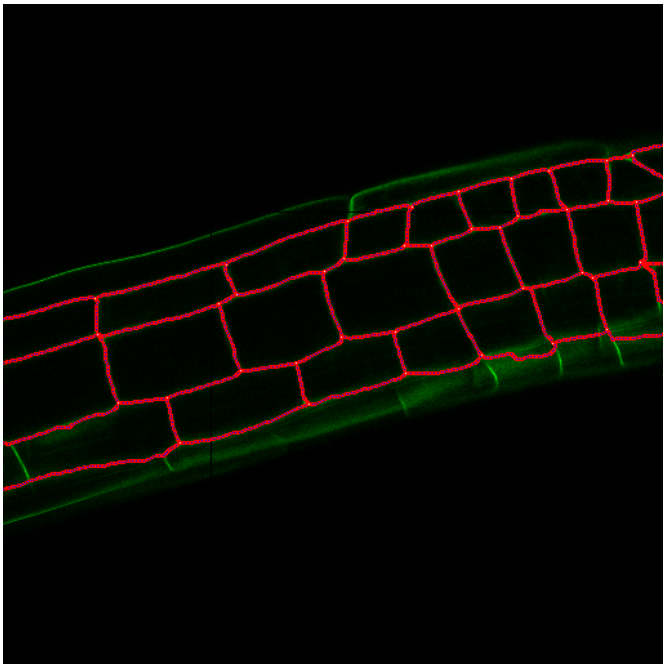
Y axis (with a negligible motion along the *X* axis). A root showing gravitropism can display a sudden turn in the direction of motion which can be identified from the change in the direction of the average velocity vector of all the network nodes and this approach can be modified to extend the boundaries along both the *X* axis and *Y* axis simultaneously as needed; but images showing gravitropic root growth are not considered in this thesis.

2. The MCMC tracker predicts the locations of all the nodes of the cells including the cells that left the field of view. Though, there is no underlying image information for the network nodes that have left the image, the MCMC tracker uses the velocity vector of their last state inside the image boundary to make a prediction on the location of the nodes.

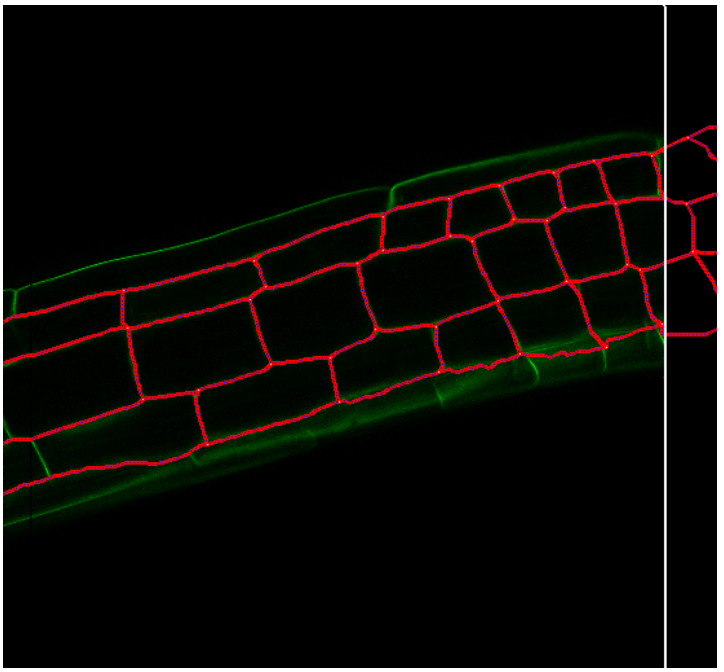
3. The predicted locations of tracked nodes are used to warp the previous network snakes using linear interpolation to fit the new locations of the network nodes. During the warping process, the network is made to remain connected to the stretched image boundary, but the shape of the cells that left the actual image boundary is preserved. During energy minimization of the network snakes, the curvature and continuity energy of the network snakes control the smoothness of the cell walls.

The prediction of the location of the cells that left the image boundary is of course a "best guess" with no underlying image information outside the image boundary, but allowing cells that have left the image to remain connected to the network and preserves the network topology.

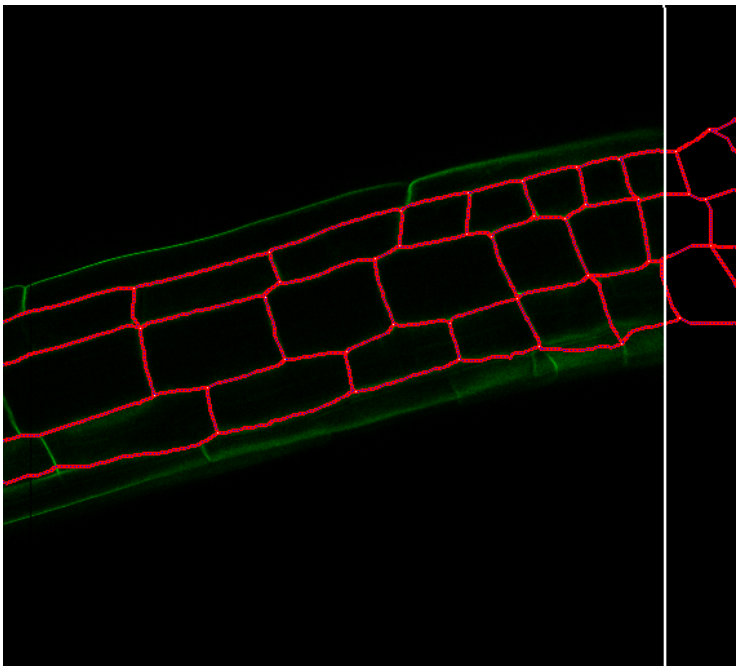
An example output shown in Figure 43, depicts the tracking of cells leaving the image, in which the white line marks the actual image boundary.



a) t=0



b) t=3



c) t=4

Figure 43. Tracking of Cells leaving the image boundary
White line marker is the actual image boundary, and the portion of the network outside this white line is the region that has already left the image, but that remains connected to the network.

7.2. Cells entering the image boundary

As cells leaving the image remain connected to the network, the topology is preserved using the approach described above. Cells entering the field of view will, however, also affect the network topology, increasing the size of the network. Newly visible cells should be included into the tracker data structure and the network topology as and when they appear. This is achieved in two steps, the detection and the subsequent tracking of the new cells.

A node detector has been implemented to detect newly-entered nodes and these new nodes are included into the node tracker using a Reversible Jump MCMC framework (RJMCMC) [24], which can cope with the change in dimensionality caused by varying numbers of targets.

While new nodes are detected, it is desirable to redefine the network snakes topology before adding them into the nodes tracker so that only nodes that form new *connectable cell boundary snakes* that fit into the current topology are added into the node tracker. For a snake to be a *connectable cell boundary* and to be able to fit into the current topology, both the ends of the snake should be connected to the existing network snakes. In what follows, detection of new nodes by forming *connectable cell boundary snakes* and tracking the newly detected nodes are explained in detail.

7.2.1. Detection of newly entered cells

To understand this implementation, a network snake should be assumed to be a network with an endless boundary; the part of the network that is currently visible is only the region inside the image boundary. So, the new cells that are yet to enter the image boundary are also connected to the network but have not yet appeared inside the image boundary. This is similar to the leaving cells that remain connected to the network but have left the image boundary. So, any entering cell must be connected to at least one of the cells in the existing network that are connected to the image boundary as shown in Figure 44.

It takes a few time steps for a cell to fully enter the image. At an intermediate time point, the cell walls and nodes of the partially visible cells are detected and added to the network. This involves the following steps

- Detection of newly entered nodes
- Forming connectable cell boundary snakes

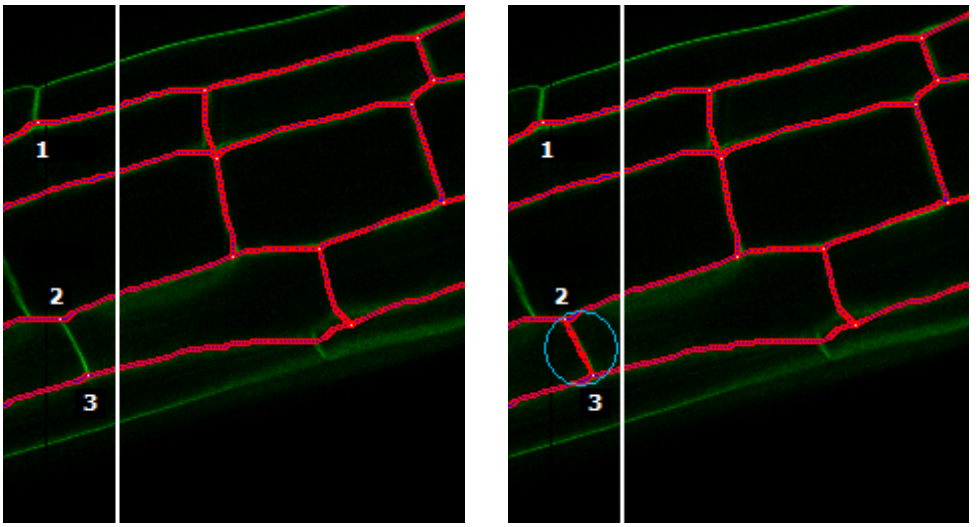


Figure 44. New cell boundary snakes detection
(a) Nodes Detection (b) Forming connectable snakes (portion of the image left to the white line marker is the newly appeared candidate region and the new set of detected nodes in this region are numbered as 1, 2 and 3, of which node-pair 2-3 forms a connectable snake)

7.2.1.1. Detection of newly entered nodes

A node detector is implemented using the junction detector model explained in section 3.4.2. Depending upon the direction of motion of the cells and the average velocity vector, a region of the image where new data can appear, known as the candidate region is identified. For example, if the cells exhibit a linear motion towards the right (positive along the *X* direction) and the average velocity of all the nodes is 15 pixels per time step along the *X* direction, then the leftmost 15 pixel-wide region in the image is chosen as the candidate region.

As noted earlier, any new cell which has entered the image boundary must be connected to the existing network snakes; the cell walls of these new cells form new nodes at the connecting points. The detector searches for these new nodes in the candidate region along the snakes in the network that are connected to the image boundary. The junction detector model (section 3.4.2) is applied for all the snake

points along the network snakes in the candidate region. Any snake point which shows a very high likelihood (above an empirically-determined threshold) of being a node is marked as a node candidate, but if this candidate is in very close proximity (below a fixed distance) to another candidate, the best of the two is chosen. This process is applied to all the snakes in the candidate region.

The image in Figure 44 shows the detection of newly entered nodes in which the newly appearing region is the portion of the image to the left of the white line marker and the new set of detected nodes in this region are numbered *1*, *2* and *3*.

7.2.1.2. Forming connectable cell boundary snakes

The detected nodes are then filtered for node-pairs that form connectable snakes. Any node that can be added to the tracker should have a matching node such that the pair when connected forms a valid snake over the underlying cell boundary, as follows.

Firstly, different combinations of node-pairs are formed from the detected nodes. Only node-pairs satisfying the following two conditions are considered to form connectable snakes.

1. Nodes in a pair should not belong to the same existing snake.

2. A node should only be paired with another node that belongs to the nearest snake connected to the image boundary.

For each node-pair, the shortest route in the network by which the nodes in the node-pair can be connected is marked. If the shortest route meets a boundary snake which does not belong to any of the nodes in the pair, the node-pair is rejected. For example in Figure 44, the node-pair 2-3 forms a connectable pair whereas node-pairs 1-2 and 1-3 cannot form a connectable pair.

Once valid connectable node-pairs are formed, they are used to form snakes. Snakes are formed from different combinations of valid node-pairs by creating snake points along a straight line between the nodes in the pair. The average image energy for a snake is calculated by taking the average of the image energies at all the snake points in the snake. The average image energy of the snake should be above a threshold image energy for the snake to be considered connectable. The threshold chosen is a local threshold which is a histogram mean over a local window as explained in section 3.4.2.1.

The set of non-repeated nodes k_d obtained from the node-pairs forming the connectable snakes are then added to the node tracker.

7.2.2. Tracking detected nodes

The existing MCMC tracking framework (section 3.5.1) used to track a *fixed* number of nodes is modified to handle *variable* number of targets. This is achieved by implementing the RJMCMC [24] framework which can cope with the variable aspect to the number of targets. The RJMCMC implementation is similar to that in [24], but this thesis describes the algorithm and the proposal acceptance ratios in more detail. To handle the problem at hand, three proposal types were chosen namely the add proposal, the delete proposal and the update proposal.

Refer to section 3.5.1 for the MCMC based particle filter algorithm, where the targets state space is represented as a set of *N* un-weighted samples X_t. A state space in RJMCMC is represented as (k_t,X_t) which contains an additional variable k_t: a set of identifiers of targets that are currently in view. In other words, each joint state sample 'i' contains the identifier k_i, a set of the targets represented by that sample.

The RJMCMC-based particle filter algorithm iterates over the following steps [24]:

1. At time *t-1*, the targets' state is represented by a set of *N* un-weighted samples $\{k_{t-1}^{r}, X_{t-1}^{r}\}_{r=1}^{N}$. Each sample 'i' is a joint state sample representing k_{t-1}^{i} set of targets.

2. Initialize the MCMC sampler by selecting a random joint state sample (k_t, X_t) from the set of *N* un-weighted samples $\{k_{t-1}^{r}, X_{t-1}^{r}\}_{r=1}^{N}$ at time *t-1*, apply the motion model to all the targets in the set k_t and calculate the likelihood P for the targets at the new positions obtained after applying the motion model.

3. Perform (B+MN) iterations to obtain *N* joint state samples. Discard the first *B* samples for burn-in and obtain the remaining *N* samples $\{k_t^{r}, X_t^{r}\}_{r=1}^{N}$ which represent the targets' state at time t.

   Propose a new state $(k_t^{'}, X_t^{'})$ after making a random proposal and calculate the acceptance ratio for the proposal as below.

   a. Add

   Randomly choose a target *i* with a uniform probability $p = (1/|k_d \setminus k_t|)$ and propose to add it to the current state, where $\{k_d \setminus k_t\}$ is the set of targets that are detected but not yet added to the current state. Compute the acceptance for the proposal using

$$A_a = \frac{P(Z_t \mid X'_t)}{P(Z_t \mid X_t)} \times \frac{p_d}{p_a} \times \frac{q_d}{q_a} \tag{24}$$

where, the first term $\frac{P(Z_t \mid X'_t)}{P(Z_t \mid X_t)}$ is the likelihood ratio, which is the ratio of the likelihood of the proposed state to the likelihood of the current state. For the add proposal, the target which it is proposed to add to the current state would not have existed in the state prior to adding and hence, the prior distribution for the target to be added is unknown. For an unknown distribution, as a best guess the density is assumed to be uniform. i.e., any value of the image measurement between *0-1* is equally likely. Hence, by using the probability density function for the uniform distribution, the likelihood of the current state prior to adding the target $P(Z_t \mid X_t)$ in the absence of a possible image measurement is given as

$$P(Z_t \mid X_t) = \frac{1}{C} \tag{25}$$

where, C is the length of the interval of the uniform distribution ranging from *a* to *b*, in this case it is 0 to 1 and hence *C = b-a = 1-0 =1* and the acceptance ratio for add proposal becomes

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{q_d}{q_a} \tag{26}$$

where,

$q_a = 1/\left| \mathrm{k_d} \setminus \mathrm{k_t} \right|$ is the probability of selecting target *i* to add to the current state

$q_d = 1/\left| \mathrm{k'_t} \cap \mathrm{k_d} \right|$ is the probability of selecting target *i* to delete from the proposed state

p_d is the probability of occurrence of the Delete proposal type

p_a is the probability of occurrence of the Add proposal type

$P(Z_t \mid X'_t)$ is the likelihood of the proposed state which is explained later in this section

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{\left| \mathrm{k_d} \setminus \mathrm{k_t} \right|}{\left| \mathrm{k'_t} \cap \mathrm{k_d} \right|} \tag{27}$$

b. Delete

Randomly choose a target *i* with a uniform probability $p = (1/\left| \mathrm{k_t} \cap \mathrm{k_d} \right|)$ and propose to delete it from the current state, where $\{\mathrm{k_t} \cap \mathrm{k_d}\}$ is the set of detected targets that were already added to the current state. Compute the acceptance for the proposal as

$$A_d = \frac{P(Z_t \mid X'_t)}{P(Z_t \mid X_t)} \times \frac{p_a}{p_d} \times \frac{q_a}{q_d} \tag{28}$$

where, the first term $\frac{P(Z_t \mid X'_t)}{P(Z_t \mid X_t)}$ is the likelihood ratio, which is the ratio of the likelihood of the proposed state to the likelihood of the current state. For the delete proposal, the target which is proposed to delete from the current state will not exist in the proposed state and hence, the proposed distribution for the deleted target is unknown. Again, for an unknown distribution, the density is assumed to be uniform. i.e., any value of the image measurement between *0-1* is equally likely. Hence, by using the probability density

function for the uniform distribution, the likelihood of the proposed state $P(Z_t \mid X'_t)$ is given as

$$P(Z_t \mid X'_t) = \frac{1}{C} \tag{29}$$

where, C is the length of the interval of the uniform distribution ranging from *a* to *b*, in this case it is 0 to 1 and hence *C = b-a = 1-0 =1* and the acceptance ratio for add proposal becomes

$$A_d = \frac{1}{P(Z_t \mid X_t)} \times \frac{p_a}{p_d} \times \frac{q_a}{q_d} \tag{30}$$

where,

$q_d = 1/\left| \mathrm{k_t} \cap \mathrm{k_d} \right|$ is the probability of selecting target *i* to delete from the current state

$q_a = 1/\left| \mathrm{k_d} \setminus \mathrm{k'_t} \right|$ is the probability of selecting target *i* to add to the proposed state

p_d is the probability of occurrence of the Delete proposal type

p_a is the probability of occurrence of the Add proposal type

$P(Z_t \mid X_t)$ is the likelihood of the current state which is explained later in this section

$$A_d = \frac{1}{P(Z_t \mid X_t)} \times \frac{p_a}{p_d} \times \frac{\left| \mathrm{k_t} \cap \mathrm{k_d} \right|}{\left| \mathrm{k_d} \setminus \mathrm{k'_t} \right|} \tag{31}$$

c. Update

This proposal keeps the dimensionality unchanged and is same as the basic MCMC sampling. Calculate the acceptance ratio as

$$A_u = \frac{P(Z_t \mid X'_t)}{P(Z_t \mid X_t)} \tag{32}$$

which is the ratio between the likelihood of the proposed state and the likelihood of the current state.

In the acceptance ratios, the likelihood term $P(Z_t \mid X_t)$ is the probability we would have observed the measurement Z_t, given the state X_t. This is obtained by evaluating the likelihood probability using a Gaussian function as defined [109] below.

$$P(Z_t \mid X_t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{33}$$

If the acceptance ratio for a proposal *m*, $A_m >= 1$, accept the proposed state. Otherwise, also accept the proposed state with probability A_m.

Ignore the first B samples for burn in, and thereafter, add a copy of X_t' to the current set $\{X_t\}$.

The resulting sample set $\{k_t^r, X_t^r\}_{r=1}^N$ forms the approximation for the current state at time *t*.

For all the proposal types defined in the RJMCMC algorithm above, the Jacobian determinant term [24] [101] [124] in the acceptance ratio is ignored as the Jacobian term is reduced to unity for the chosen proposal types [24] [101].

7.2.3. Implementation of Reversible Jump MCMC

Each move in RJMCMC should have a corresponding reverse move defined. For example, a birth move should have a corresponding death move. In this implementation, for the add proposal which increases the state space dimensionality by 1, the reverse move is the delete proposal that decreases the dimensionality by 1. The reverse move for the update proposal which keeps the dimensionality unchanged is the update proposal itself.

Proposal types are carefully chosen such that the transition from current state $\{X_t\}$ to the proposed state $\{X'_t\}$ for any move type satisfies the three conditions below [101].

1. The transition should be a bijection function that forms a one-to-one correspondence between states.

2. Its derivative should be invertible with a nonzero Jacobian determinant. For the chosen move types, the Jacobian determinant reduces to unity [24] [101].

3. It has a corresponding reverse move that can be applied to recover the original state and satisfies the above two conditions.

Consider the example shown in Figure 45 (a), where nodes 1,2,3 and 4 are newly detected nodes of which only node-pairs 2-3 and 3-4 are connectable pairs.

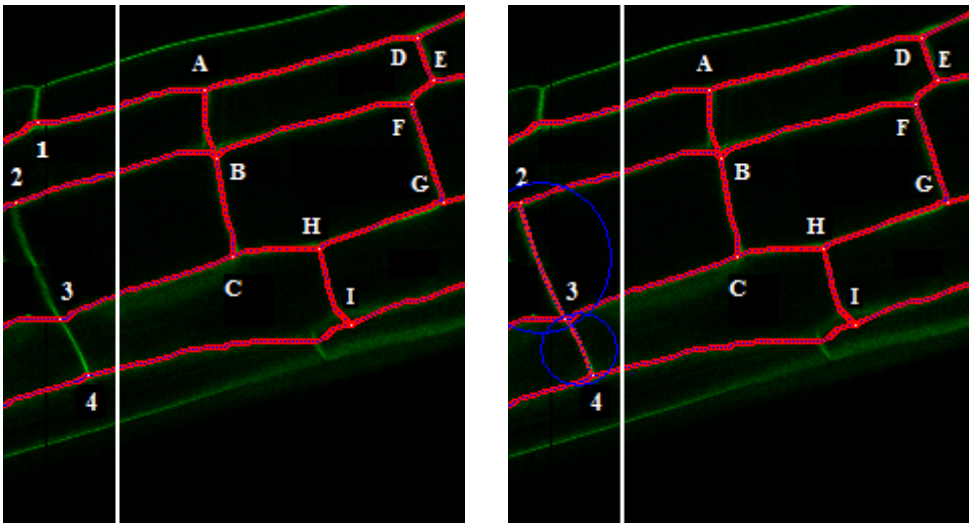


Figure 45. New cell boundary snakes detection and tracking
(a) Nodes Detection (portion of the image left to the white line marker is the newly appeared region and the new set of detected nodes in this region are numbered as 1, 2 ,3 and 4 (b) Connectable snakes (node-pair 2-3 and 3-4 forms connectable snakes) to be included in the tracker's state

Nodes $\{A,B,C,D,E,F,G,H,I\}$ with dimension *9* are present in the current state $\{X_t\}$ and the set of detected targets $\{K_d\}$ is $\{2,3,4\}$ with dimension *3* and likelihood values of {1.2, 1.0, 1.3} respectively. Detected node *1* is excluded because it cannot form a valid pair with any other detected node. RJMCMC should be implemented in the tracker aiming to include the node-pairs 2-3 and 3-4 into the tracker as shown in Figure 45 (b). To this, several options exist.

There are three possible ways of implementing RJMCMC within the NS-MCMC tracker:

1. Treating node-pairs as paired target objects
2. Treating *individual* nodes as target objects
3. Treating individual nodes as target objects with an intelligent object detector controlling the construction of the tracker state.

In what follows, all three possible ways are discussed in detail leading to a conclusion justifying the chosen approach.

7.2.3.1. Treating node-pairs as target objects for the RJMCMC tracker

In this approach, node-pairs that form cell boundaries are treated as target objects so that nodes are added to the tracker in pairs. This approach has several complexities to deal with.

Consider the example in Figure 45, from which the set of detected targets $\{K_d\}$ would have a set of node-pairs {2-3, 3-4} and the current state $\{X_t\}$ contains the set of nodes {*A,B,C,D,E,F,G,H,I*}. Consider the below scenario.

1. In an add proposal, node-pair *2-3* is randomly chosen and added to the current state so that the proposed state $\{X'_t\}$ becomes {*A,B,C,D,E,F,G,H,I,**2,3***} increasing the dimensionality by ***two***. Assuming the proposal is accepted, the current state $\{X_t\}$ becomes {*A,B,C,D,E,F,G,H,I,**2,3***} with dimension *11*.

2. In another add proposal, node-pair *3-4* is chosen and proposed to add it to the current state. But node *3* is already present in the current state after the step 1 above and hence to avoid the repetition of targets, only node *4* can now be added. So, the proposed state $\{X'_t\}$ becomes {*A,B,C,D,E,F,G,H,I,**2,3,4***} increasing the dimensionality by ***one***.

In this scenario, the first add proposal increases the state space dimensionality by ***two*** whereas the second add proposal increases the dimensionality by ***one*** which means that the dimensionality changes in the add proposal steps are not consistent. For this add proposal that defines a transition from current state $\{X_t\}$ to the proposed state $\{X'_t\}$ with inconsistent dimensionality changes, defining a corresponding reverse proposal satisfying the three conditions given above is a complex task.

7.2.3.2. Treating individual nodes as target objects

In this approach, individual nodes are treated as target objects. So, for the example in Figure 45, the set of detected targets $\{K_d\}$ would have a set of individual nodes {2,3,4} with likelihood values of {1.2, 1.0, 1.3} respectively and the current state $\{X_t\}$ contains the set of nodes $\{A,B,C,D,E,F,G,H,I\}$. The inconsistent dimensionality of move types in the previous approach of considering node-pairs as target objects is solved in this approach. The add proposal always increases the dimensionality by one and the corresponding reverse proposal, the delete proposal decreases the dimensionality by one and the moves also satisfy the three conditions stated earlier in the section.

Consider the following example steps in RJMCMC sampling where k_t is the current state and k'_t is the proposed state. Let us choose the probability of selecting the add proposal p_a = probability of selecting delete proposal p_d, so that the ratio

$$\frac{p_a}{p_d} = \frac{p_d}{p_a} = 1$$

Move 1.Add proposal: Propose to add node 2 to the state

$k_t = \{A,B,C,D,E,F,G,H,I\}$ $\quad k'_t = \{A,B,C,D,E,F,G,H,I,\underline{\mathbf{2}}\}$

$k'_t \cap k_d = \{2\}$ $\qquad k_d \setminus k_t = \{3,4\}$

From Eq. (27), acceptance ratio is

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{|k_d \setminus k_t|}{|k'_t \cap k_d|}$$

$$= 1.2 \times 1 \times \frac{2}{1} = 2.4 > 1$$

Proposal accepted.

Move 2. Delete proposal: Propose to delete node 2 from the state

$\mathrm{k}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2}}\}$ $\quad \mathrm{k'}_t = \{A,B,C,D,E,F,G,H,I\}$

$\mathrm{k}_t \cap \mathrm{k}_d = \{2\}$ $\qquad \mathrm{k}_d \setminus \mathrm{k'}_t = \{2,3,4\}$

From Eq. (31), acceptance ratio is

$$A_d = \frac{1}{P(Z_t \mid X'_t)} \times \frac{p_a}{p_d} \times \frac{\left| k_t \cap k_d \right|}{\left| k_d \backslash k'_t \right|}$$

$$= \frac{1}{1.2} \times 1 \times \frac{1}{3} = 0.278 < 1$$

Proposal (most likely) rejected.

Move 3. Add proposal: Propose to add node 3 to the state

$\mathrm{k}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2}}\}$ $\quad \mathrm{k'}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2,3}}\}$

$\mathrm{k'}_t \cap \mathrm{k}_d = \{2,3\}$ $\qquad \mathrm{k}_d \setminus \mathrm{k}_t = \{4\}$

From Eq. (27), acceptance ratio is

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{\left| k_d \backslash k_t \right|}{\left| k'_t \cap k_d \right|}$$

$$= 1.0 \times 1 \times \frac{1}{2} = 0.5 < 1$$

Proposal accepted with a random probability *p=0.5*.

Move 4. Delete proposal: Propose to delete node 3 from the state

k_t = {*A,B,C,D,E,F,G,H,I,**2,3***} k'_t = {*A,B,C,D,E,F,G,H,I,**2***}

$k_t \cap k_d$ = {*2,3*} $k_d \setminus k'_t$ = {*3,4*}

From Eq. (31), acceptance ratio is

$$A_d = \frac{1}{P(Z_t \mid X'_t)} \times \frac{p_a}{p_d} \times \frac{|k_t \cap k_d|}{|k_d \setminus k'_t|}$$

$$= \frac{1}{1.0} \times 1 \times \frac{2}{2} = 1$$

Proposal accepted.

Move 5. Add proposal: Propose to add node 4 to the state

k_t = {*A,B,C,D,E,F,G,H,I,**2***} k'_t = {*A,B,C,D,E,F,G,H,I,**2,4***}

$k'_t \cap k_d$ = {*2,4*} $k_d \setminus k_t$ = {*3*}

From Eq. (27), acceptance ratio is

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{|k_d \setminus k_t|}{|k'_t \cap k_d|}$$

$$= 1.3 \times 1 \times \frac{1}{2} = 0.65 < 1$$

Proposal accepted with a random probability *p=0.65*.

From the above steps in RJMCMC sampling, the following key points should be noted.

1. Acceptance of add proposals decreases with the increase in number of targets added to the current state, so that it is more likely to reject the add proposals after adding a few targets to the state. This is because the acceptance of the add proposal is directly proportional to $\left|k_d \setminus k_t\right|$, the number of detected targets that were not yet added to the current state and is inversely proportional to $\left|k'_t \cap k_d\right|$, the number of detected targets that were already added to the current state.

2. Acceptance of delete proposals increases with the increase in number of targets added to the current state so that it is more likely to accept the delete proposals after adding a few targets to the state. This is because the acceptance of the delete proposal is directly proportional to $\left|k_t \cap k_d\right|$, the number of detected targets that were already added to the current state and is inversely proportional to $\left|k_d \setminus k'_t\right|$, the number of detected targets that were not yet added to the current state.

From the above points, it is clear that the acceptance ratios for the reversible moves in RJMCMC sampling are designed in such a way that the dimensionality of the state space at time *t* is always less than the sum of the dimensionality of the state space at time *t-1* and the number of detected targets at time *t*. i.e., all the detected targets $\{K_d\}$ cannot be added to the tracker at one time. This approach is useful for detectors that contain several False Positives where this approach only adds the true targets with high likelihood values leaving behind the targets with low likelihood values which are more likely to be False Positives. This partial inclusion of detected targets is fine for tracking frameworks that track distinct targets such as [24] [101], but for the

application considered in this thesis which involves a network of connected cells, this is problematic, as explained below.

Consider the state $\{X_t\}$ after *Move 5* above which represents the set of targets k_t = {*A,B,C,D,E,F,G,H,I,**2,4***} with detected nodes *2* and *4* added into the current state. Assume, this happens to be the modal configuration for the resulting *N* joint state samples of varying dimensions so that nodes *2* and *4* are added to the tracker at time *t*. But from Figure 45, nodes *2* and *4* do not form a connectable snake in the network and without node *3*, the nodes *2* and *4* are not considered as network nodes. Nodes *2* and *4* are treated as network nodes only when node *3* is present so that node-pairs 2-3 and 3-4 form valid network snakes. Moreover, at the detection stage as explained in section 7.2.1, the network topology should have redefined to include the newly detected connectable snakes. The exclusion of node *3* in the tracker's data structure orphans the newly included connectible snakes *2-3* and *3-4* and destroys the redefined topology which is undesirable. To avoid this undesirable effect, the tracker should include all the valid detected nodes as explained in the next approach to achieve the configuration in Figure 45 (b).

7.2.3.3. Treating individual nodes as target objects for RJMCMC tracker with an intelligent object detector controlling the construction of the tracker state

In this tracking framework, the detector that detects the newly entered cells is intelligent enough to do all the necessary filtering and provides a set of detected nodes that show a strong likelihood for their existence and importantly can fit into the network topology following the set of rules explained in section 7.2.1. An actual image measurement is used to obtain the likelihood for the nodes detection, but the detected nodes that have passed the filtering process and form connectable snakes are believed to have the highest possible likelihood values in the likelihood distribution.

In other words, the detector is considered very reliable. So, these nodes are added to the tracker with the highest possible likelihood. With a high likelihood value, all the detected nodes are more likely to get included into tracker's state as shown below.

Assuming the measurement of the state density is distributed normally, the highest possible likelihood or the peak of the distribution is determined as follows. The PDF for the normal distribution is given as [109]

$$P(Z_t \mid X_t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{34}$$

The highest possible likelihood occurs at the mean value i.e., $x = \mu$ which is the peak of the Gaussian curve. From an empirical study on the likelihood measurements from the images considered in this thesis, the standard deviation σ is found to be 0.2. The highest possible likelihood is determined from Eq. (34) as below.

$$P(Z_t \mid X_t)_{Max} = \frac{1}{\sqrt{2\pi(0.2)^2}} e^{-\frac{(\mu-\mu)^2}{2\sigma^2}}, \text{ as } x = \mu$$

$$= \frac{1}{\sqrt{2\pi(0.2)^2}} e^{0}$$

$$= 1.995$$

Now consider the RJMCMC sampling below with the same moves as in the previous approach (section 7.2.3.2) where $k_t = \{A,B,C,D,E,F,G,H,I\}$ is the current state, $\mathrm{k'_t}$ is the proposed state and the set of detected targets $k_d = \{2,3,4\}$. All the detected nodes now have the highest possible likelihood value ***1.995***.

Move 1. Add proposal: Propose to add node 2 to the state

$\mathrm{k}_t = \{A,B,C,D,E,F,G,H,I\}$ $\quad \mathrm{k'}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2}}\}$

$\mathrm{k'}_t \cap \mathrm{k}_d = \{2\}$ $\qquad \mathrm{k}_d \setminus \mathrm{k}_t = \{3,4\}$

From Eq. (27), acceptance ratio is

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{\left| k_d \backslash k_t \right|}{\left| k'_t \cap k_d \right|}$$

$$= 1.995 \times 1 \times \frac{2}{1} = 3.99 > 1$$

Proposal accepted.

Move 2. Delete proposal: Propose to delete node 2 from the state

$\mathrm{k}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2}}\}$ $\quad \mathrm{k'}_t = \{A,B,C,D,E,F,G,H,I\}$

$\mathrm{k}_t \cap \mathrm{k}_d = \{2\}$ $\qquad \mathrm{k}_d \setminus \mathrm{k'}_t = \{2,3,4\}$

From Eq. (31), acceptance ratio is

$$A_d = \frac{1}{P(Z_t \mid X'_t)} \times \frac{p_a}{p_d} \times \frac{\left| k_t \cap k_d \right|}{\left| k_d \backslash k'_t \right|}$$

$$= \frac{1}{1.995} \times 1 \times \frac{1}{3} = 0.167 < 1$$

Proposal (most likely) rejected.

Move 3. Add proposal: Propose to add node 3 to the state

$\mathrm{k}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2}}\}$ $\quad \mathrm{k'}_t = \{A,B,C,D,E,F,G,H,I,\underline{\boldsymbol{2,3}}\}$

$k'_t \cap k_d = \{2,3\}$ $\quad$ $k_d \setminus k_t = \{4\}$

From Eq. (27), acceptance ratio is

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{|k_d \setminus k_t|}{|k'_t \cap k_d|}$$

$$= 1.995 \times 1 \times \frac{1}{2} = 0.9975 \simeq 1$$

Proposal accepted with a random probability *p=0.9975*.

Move 4. Delete proposal: Propose to delete node 3 from the state

$k_t = \{A,B,C,D,E,F,G,H,I,\underline{\mathbf{2,3}}\}$ $\quad$ $k'_t = \{A,B,C,D,E,F,G,H,I,\underline{\mathbf{2}}\}$

$k_t \cap k_d = \{2,3\}$ $\quad$ $k_d \setminus k'_t = \{3,4\}$

From Eq. (31), acceptance ratio is

$$A_d = \frac{1}{P(Z_t \mid X'_t)} \times \frac{p_a}{p_d} \times \frac{|k_t \cap k_d|}{|k_d \setminus k'_t|}$$

$$= \frac{1}{1.995} \times 1 \times \frac{2}{2} = 0.5 < 1$$

Proposal rejected[1].

Move 5. Add proposal: Propose to add node 4 to the state

$k_t = \{A,B,C,D,E,F,G,H,I,\underline{\mathbf{2,3}}\}$ $\quad$ $k'_t = \{A,B,C,D,E,F,G,H,I,\underline{\mathbf{2,3,4}}\}$

$k'_t \cap k_d = \{2,3,4\}$ $\quad$ $k_d \setminus k_t = \{4\}$

[1] This delete proposal is equally likely to be accepted with a random probability. If it is accepted, the subsequent add proposal will have a high acceptance ratio.

From Eq. (27), acceptance ratio is

$$A_a = P(Z_t \mid X'_t) \times \frac{p_d}{p_a} \times \frac{|k_d \backslash k_t|}{|k'_t \cap k_d|}$$

$$= 1.995 \times 1 \times \frac{1}{3} = 0.67 < 1$$

Proposal (mostly likely) accepted with a random probability *p=0.67*.

The state $\{X_t\}$ achieved after *Move 5* above with set of targets k_t = {*A,B,C,D,E,F,G,H,I,**2,3,4***} successfully includes all the detected targets {*2,3,4*} unlike the previous approach where node *3* was missed destroying the network topology. Thus, by increasing trust in our intelligent object detector and using the highest possible likelihood for the detected nodes and thereby increasing the chance of acceptance of add proposals during the construction of Markov chain, the desired network configuration shown in Figure 45 is achieved.

Some examples of tracking variable number of targets using this approach are shown in Figure 46 and Figure 47.

7.2.4. Estimating the location of targets from the samples of varying dimensions

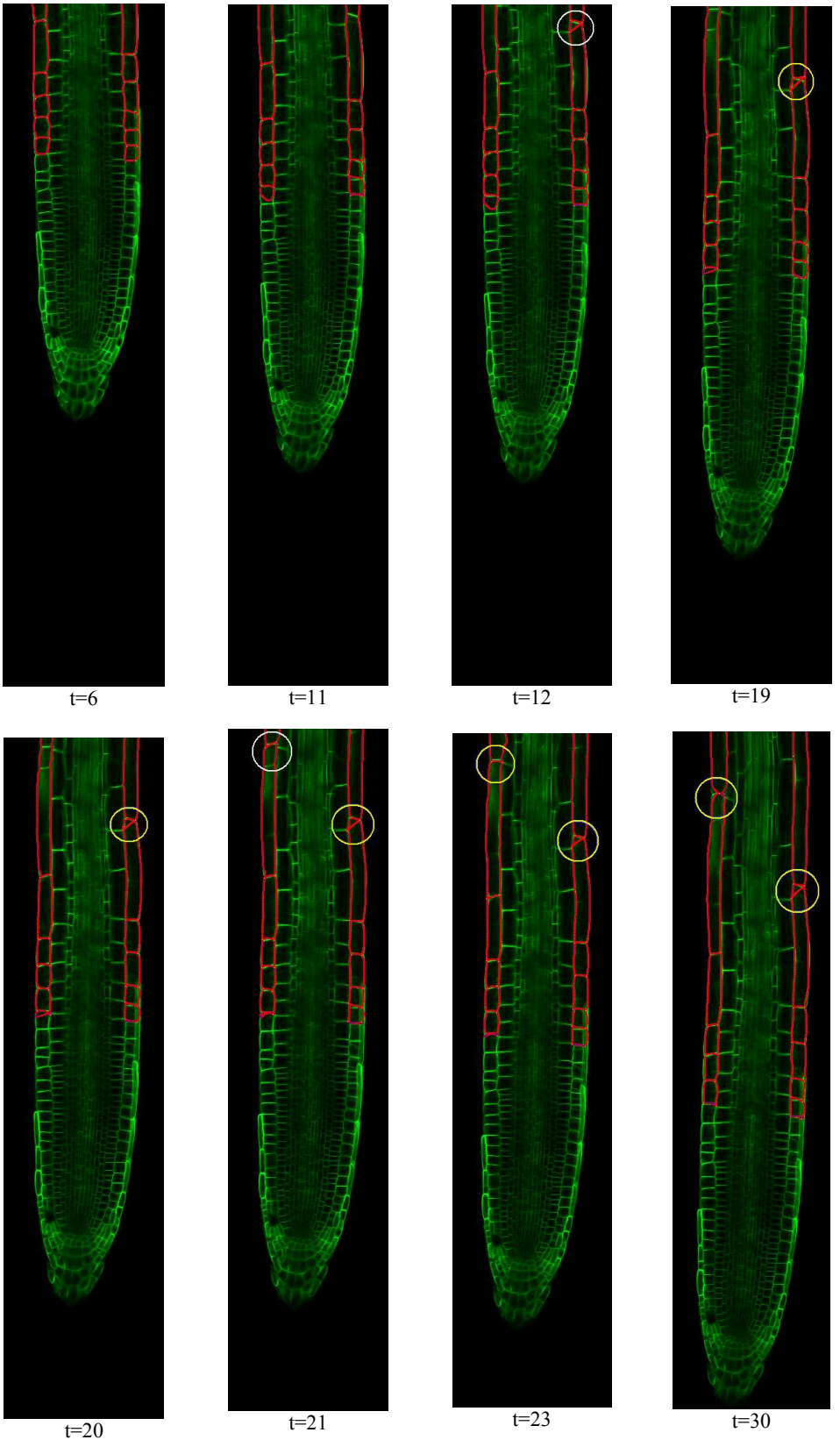
Taking a mean over all the samples has been a good solution to estimate the location of targets for the tracker. This is, however, a feasible solution only if tracking involves a fixed number of targets, in which the sample set representing the state space contains samples with the same dimensionality. To handle variable numbers of targets, RJMCMC sampling is used which constructs a Markov chain returning a set of samples with varying dimensionality. Here, each joint state sample represents a set of targets which may be different in cardinality from other samples; For example, in

a sample set {S}, a joint state sample S_i might represent targets *A*, *B* and *C* with dimension *3* and another joint state sample S_j might represent targets *A*, *C, D* and *E* with dimension *4* and others may be of different dimensions for which a mean cannot be computed. To estimate the best location of targets from this state space distribution of varying dimensionality, find a mode configuration across the set of identifiers (not just the dimension i.e., a sample representing targets *A* and *B* with dimension *2* is not the same as another sample with the same dimension but representing a different set of targets *A* and *C*) over all the samples. This mode configuration is the set of identifiers that has occurred the most number of times in the sample set. Then, take a mean of location of targets on all the samples that represent the mode configuration to predict the location of the targets. The predicted set of targets in the current time step is the set of identifiers in the mode configuration.

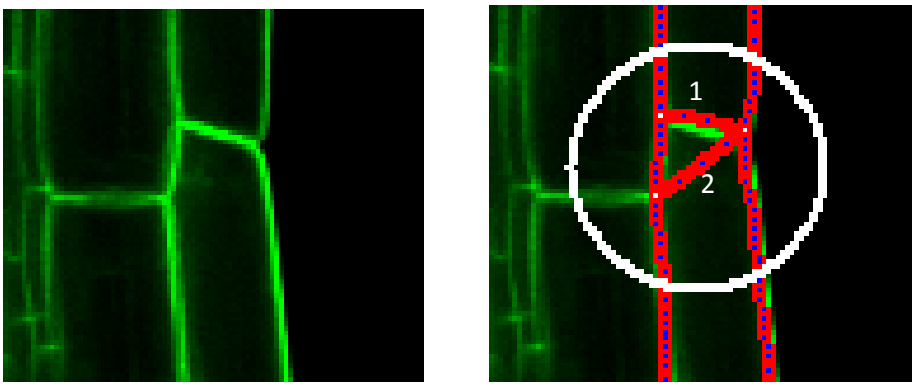
7.2.5. Results and Discussion

The approach presented has been implemented within a software tool using Visual C++.NET. Using the tool, the processing rate for a sequence of 240 x 1024 pixel images on an Intel(R) core2 duo 1.5GHz processor is approximately 7-8 Frames per minute which is fast enough for practical application given that processing speed is not a critical issue in this area.

Based on the evaluation techniques explained in chapter 5, the developed NS-RJMCMC MRF-based hybrid tracker for tracking a variable number of targets will now be evaluated on time lapse sequences of confocal images of growing Arabidopsis roots. Example frames from the results of running the NS-RJMCMC MRF-based hybrid tracker on real-life image sequences are shown in Figure 46 and Figure 47 and the corresponding evaluation metrics obtained from the tracker are presented in Table 10.



(a)



(b)

Figure 46. Tracking of entering cells over a time based sequence
(a) Note the detection (circled in white) and tracking (circled in yellow) of newly entered cell wall
(b) Cell detection at time t=12, Out of 2 cell walls detected, cell 2 is a False Positive

Figure 46 (a) shows the detection and tracking of cells entering the image in the elongation zone of the root. At time *t*=12, two new cell walls are detected (circled in white), are added to the tracker and tracked (circled in yellow) from time *t*=13. At time *t*=21, a third cell wall is detected (circled in white), is added to the tracker and tracked (circled in yellow) from *t*=22. Note at time *t*=30, three new cell walls are being tracked.

As shown in the close up view of the local image region Figure 46 (b), out of the two new cell walls detected at time *t*=12 cell wall *2* is a False Positive. The node-pair forming the cell wall *2* is qualified as a candidate for a connectable snake satisfying all the rules defined in section 7.2.1.2. This cell wall is expected to have been disqualified during thresholding based on the average image energy of the formed snake, but because of the green pixels around the snake the average image energy of the snake falls above the determined threshold (in this case, the average image energy is *95* and the threshold value is *75*) and hence is qualified as a valid connectable snake. This occurs only rarely when the new nodes appear within very

close proximity and when the underlying local image region contains a high foreground noise thereby increasing the average energy of the new cell boundary snake.

Figure 47 shows an example output of tracking that includes entering and leaving cells over a higher magnification sequence. Note the tracking of leaving cells to the right of the actual image boundary (white line) which remain connected to the extended image boundary. At time *t*=3, a new cell wall is detected (circled in white) and is added to the tracker which starts tracking (circled in yellow) the new cell wall from *t*=4.

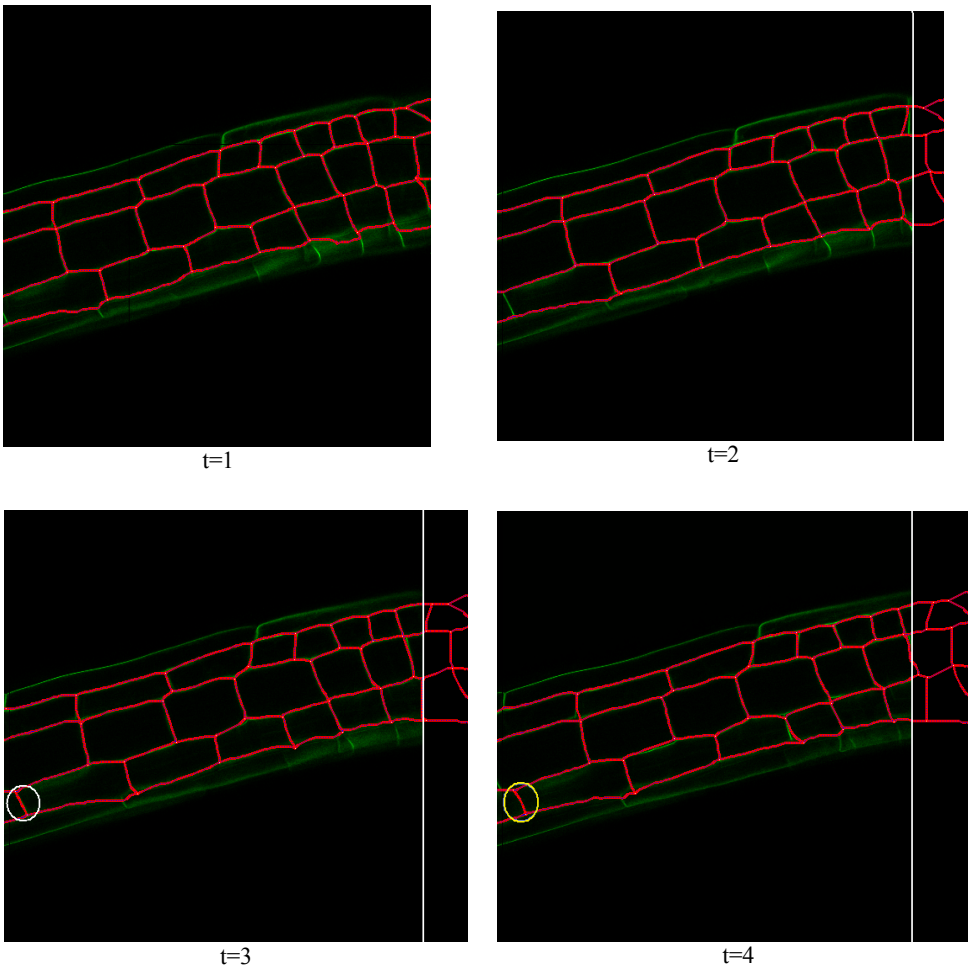


Figure 47. Tracking of entering and leaving cells over a time based sequence
Note the continued tracking of leaving cells to the right of the white line image boundary, necessary to maintain the network, and the detection (circled in white) and tracking (circled in yellow) of a newly-entered cell wall

Real Image Sequence	TRDR	FAR	OTE
Time based sequence ***1*** – Tracking cells in the elongation zone (Figure 46)	0.95	0.07	2.96
Time based sequence ***2*** (Figure 47)	0.86	0.07	2.22

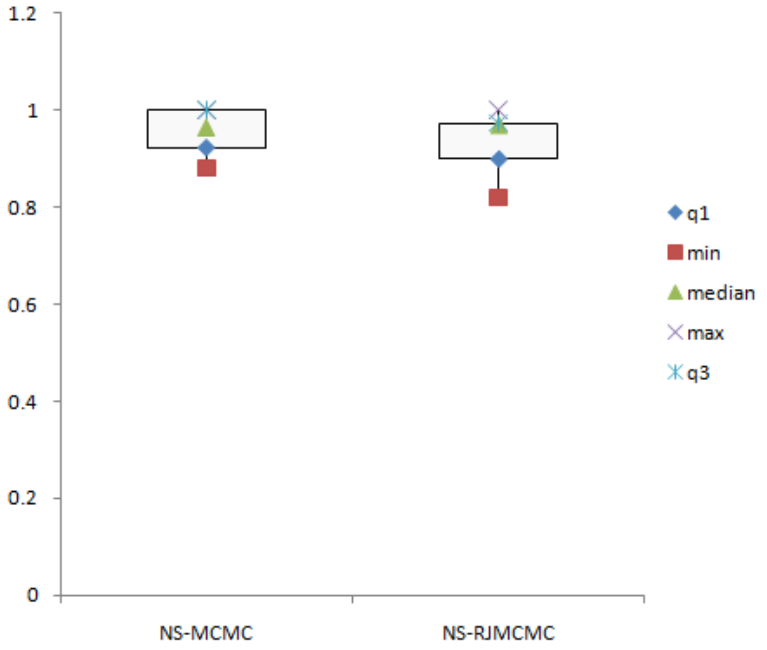
Table 10. Evaluation results on real life image sequences

As can be seen in Table 10, NS-RJMCMC MRF-based hybrid tracker performed well with a very high tracker detection rate and a low false alarm rate on the tested real image sequences. As the advantages and the limitations of the previous version of the tracker have already been discussed in chapter 6, the new NS-RJMCMC MRF based tracker for tracking variable number of targets is compared against the NS-MCMC MRF based tracker to prove that the limitations have been overcome while maintaining the accuracy of the tracker. The previous version of the tracker, the NS-MCMC MRF-based tracker is limited to fixed number of targets. But, the new NS-RJMCMC tracker can track a variable number of cells and cope well with entering and leaving cells as shown in Figure 46 and Figure 47.

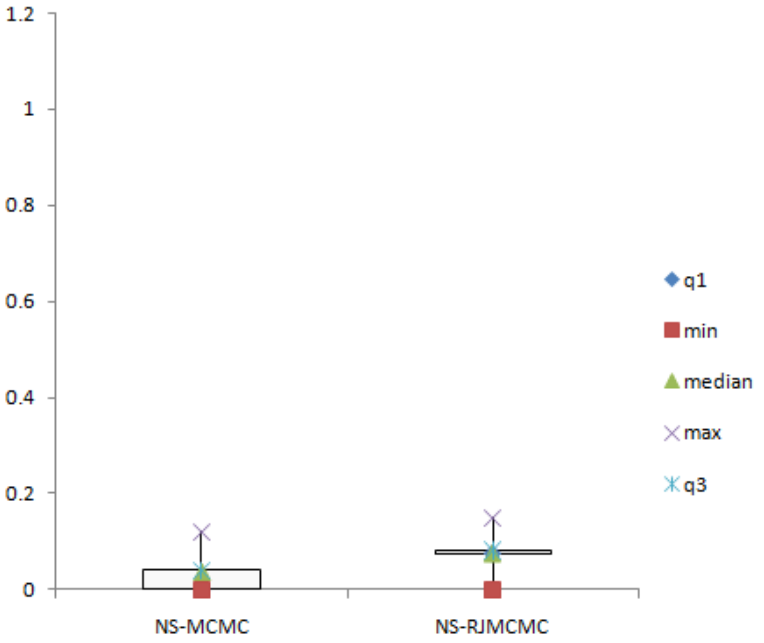
Both the NS-MCMC tracker and the NS-RJMCMC tracker are run on the first 25 frames of the sequence shown in Figure 47 and the comparison of the results is presented as box plots drawn against the performance metrics TRDR, FAR and OTE in Figure 48. The box plots for TRDR and OTE clearly show that there is no practical difference between the performance of the NS-RJMCMC tracker and the NS-MCMC tracker, though the FAR box plot with a low q1-q3 spread for the NS-RJMCMC tracker shows a small improvement compared to the NS-MCMC tracker.

All the metrics obtained at each frame for both NS-MCMC and NS-RJMCMC trackers on the sequence in Figure 47 is tabulated in Table 11 and Table 12

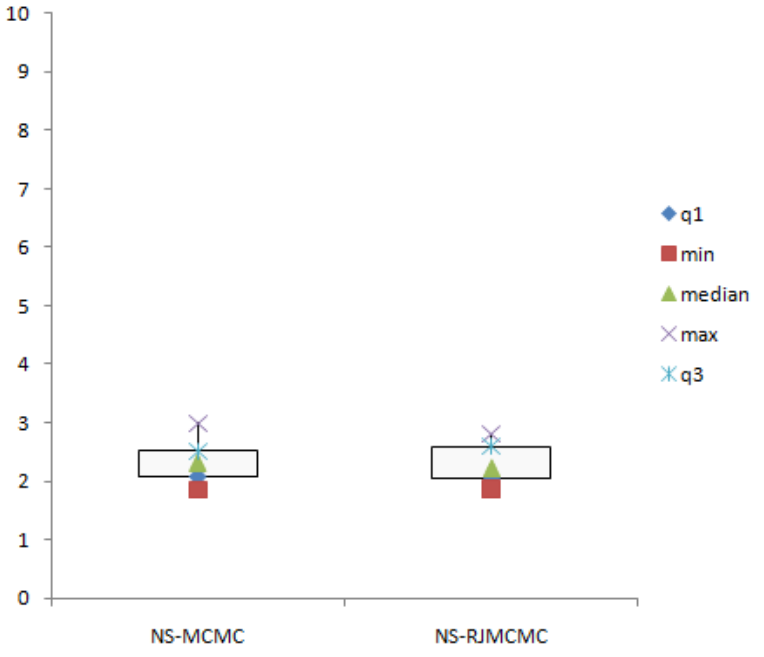
respectively. The comparison shows no major difference in the values of average TRDR, FAR and OTE between the trackers, but the NS-RJMCMC tracker shows a large number of True Positive tracks compared to the NS-MCMC tracker. This is because, during the evaluation of the previous version of the tracker, the NS-MCMC MRF based tracker is limited to a fixed number of targets. In Chapter 6, a bounding box solution was suggested to prevent the analysis of new data entering the image, focusing only on the cells that exist in the image at time *t=0*. This is no longer required as the new NS-RJMCMC tracker can cope with entering and leaving cells and the tracker can now track all the cell walls inside the image boundary; hence the large number of True Positive tracks.



(a) TRDR



(b) FAR



(c) OTE

Figure 48. Box plot comparison of NS-MCMC and NS-RJMCMC trackers

FRAME #	TP	FP	FN	TG	TT	TRDR	FAR	OTE
1	28	0	0	28	28	1.00	0.00	2.06
2	28	0	0	28	28	1.00	0.00	2.51
3	28	0	0	28	28	1.00	0.00	2.15
4	28	0	0	28	28	1.00	0.00	2.27
5	28	0	0	28	28	1.00	0.00	2.25
6	25	3	3	28	28	0.89	0.11	1.98
7	28	0	0	28	28	1.00	0.00	2.63
8	28	0	0	28	28	1.00	0.00	2.06
9	27	1	1	28	28	0.96	0.03	2.07
10	27	1	1	28	28	0.96	0.03	2.08
11	28	0	0	28	28	1.00	0.00	2.31
12	25	1	1	26	26	0.96	0.04	2.17
13	25	1	1	26	26	0.96	0.04	2.22
14	25	1	0	25	26	1.00	0.04	2.99
15	24	1	1	25	25	0.96	0.04	2.77
16	24	1	1	25	25	0.96	0.04	2.76
17	24	1	1	25	25	0.96	0.04	2.44
18	25	0	0	25	25	1.00	0.00	2.89
19	25	0	0	25	25	1.00	0.00	2.68
20	25	0	0	25	25	1.00	0.00	2.37
21	22	3	3	25	25	0.88	0.12	2.46
22	24	1	2	26	25	0.92	0.04	2.43
23	22	3	3	25	25	0.88	0.12	1.99
24	22	3	3	25	25	0.88	0.12	1.84
25	22	3	3	25	25	0.88	0.12	2.33
Average:	**25**					**0.96**	**0.04**	**2.35**

Table 11. **NS-MCMC** Computation of average TRDR, FAR and OTE for the sequence in Figure 46

FRAME #	TP	FP	FN	TG	TT	TRDR	FAR	OTE
1	32	0	0	32	32	1.00	0.00	1.94
2	32	0	1	33	32	0.97	0.00	2.41
3	32	0	0	32	32	1.00	0.00	1.95
4	32	0	0	32	32	1.00	0.00	2.15
5	32	0	2	34	32	0.94	0.00	2.12
6	30	2	5	35	32	0.86	0.06	1.99
7	34	3	1	35	37	0.97	0.08	2.03
8	34	3	1	35	37	0.97	0.08	2.00
9	34	3	2	36	37	0.94	0.08	2.25
10	34	3	1	35	37	0.97	0.08	2.22

11	33	4	2	35	37	0.94	0.11	2.13
12	33	4	2	35	37	0.94	0.11	2.04
13	34	3	1	35	37	0.97	0.08	2.07
14	34	3	4	38	37	0.89	0.08	1.86
15	34	3	4	38	37	0.89	0.08	2.75
16	32	5	7	39	37	0.82	0.13	2.47
17	37	3	2	39	40	0.94	0.07	2.65
18	37	3	1	38	40	0.97	0.07	2.22
19	37	3	1	38	40	0.97	0.07	2.71
20	37	3	1	38	40	0.97	0.07	2.29
21	37	3	1	38	40	0.97	0.07	2.73
22	36	4	4	40	40	0.90	0.10	2.64
23	37	3	1	38	40	0.97	0.07	2.59
24	37	3	2	39	40	0.95	0.07	2.79
25	34	6	4	38	40	0.89	0.15	2.29
Average:	**34**					**0.95**	**0.07**	**2.29**

Table 12. **NS-RJMCMC** Computation of average TRDR, FAR and OTE for the sequence in Figure 46

As discussed above, the NS-RJMCMC tracker is able to transform the network topology to cope with cells that are entering and leaving. However, tracking topological changes caused by motion in depth, especially when the growing root twists, is another complex task that remains unsolved and is clearly a direction for future research. Over z-stacks, due to blurring and ghosting effects, the cell boundaries from adjacent layers form significant clutter in the datasets. The appearance of these additional features is hard to predict, and many are both very faint and visible for only short time periods.

8 Conclusion and future work

8.1. Conclusion

This chapter concludes the thesis by summarising the proposed hybrid technique and discussing its performance against the goals of this thesis. Finally, unsolved problems and recommendations for future work are provided.

The main aim of this thesis is to develop a novel technique for tracking network-structured multiple deformable objects. The novel technique proposed uses a combination of the Network Snakes technique and Reversible Jump Markov Chain Monte Carlo (RJMCMC) based particle filters technique and uses a novel method based on a two level watershed algorithm for automatic initialization of the network snakes. The proposed technique is applied to a real problem of tissue level segmentation and automatic tracking of networks of cells in confocal images of the roots of the model plant *Arabidopsis thaliana*. The global structure of the connected cells can be exploited to our advantage by the high level techniques adopted here to improve upon the results obtained from lower level, local techniques such as watersheds and individual active contours.

A variant of the recently developed network snakes [22] approach to segment cells has been developed and a variant of watershed algorithm [107] has been used to

initialize the network. This approach alone is insufficient to maintain segmentation through a time series of a growing plant, in which the network can translate, elongate and warp. If such growth is sufficiently slow and/or the temporal resolution of the images is sufficiently high, then the implicit tracking ability of the snakes can be exploited to maintain cell identity over time. In many cases, however, growth between frames is sufficiently large to make this impossible; the network snakes cannot reacquire the correct boundary set when reinitialized in the same configuration on the next image. It is often not possible to increase the frequency of the image capture too much because regularly illuminating the root with the laser can have detrimental effects on growth, as well as degrading the fluorescent protein performance. The problem of maintaining the network structure over time is therefore addressed by incorporating a multi-target particle filter-based MCMC MRF tracker [23] into the network snakes framework. This tracks the locations of network junction nodes between frames, allowing the snake to be translated appropriately and warped onto the new image before its energy function is once again minimized. Being coupled with a dedicated and robust tracker, this hybrid approach can track relatively large jumps and deformations in the network, as shown in the results.

When the root exhibits normal growth, new cells enter and some existing cells leave the field of view and hence, the number of nodes and cell boundary snakes representing the network is variable. To cope with entering and leaving cells, the NS-MCMC MRF based hybrid tracker is extended to track a *variable* number of deformable objects by incorporating a Reversible Jump MCMC algorithm.

Another contribution of this thesis is a novel method for evaluating network-structured multi target tracking. This method is then used to evaluate the developed tracking framework for accuracy and robustness using real and synthetic time-lapse and 3D z-stack sequences of CLSM images of growing Arabidopsis roots with

varying degree of complexity and has been shown to be a capable method of tracking under both biologically feasibly and synthetic situations.

The algorithm performed well on all the time based and 3D z-stack sequences of CLSM images showing deformation of cells and exhibiting different types of motion modalities; namely linear, elongation and bending motions. The improvement over independent MCMC based tracking, and particularly the implicit tracking ability of snakes, is noted. The results show that the hybrid tracker performed consistently better than NS alone and MCMC alone tracker in all the tested sequences. The approach also proved to be successful at all realistic noise levels, and performed capably even under severe, unrealistic noise. As shown in the results, NS-RJMCMC tracker can track variable number of cells and cope well with entering and leaving cells.

Based on the proposed hybrid technique, a software tool has been developed to assist the extraction of growth related metrics from confocal images of growing Arabidopsis roots. This tool is being used by biologists and mathematical modellers at the Centre for Plant Integrative Biology to extract model parameters such as cell counts, cell boundaries and cell wall elongation rates from 2D images. The technique can be extended to measure parameters such as cell volumes when applied to 3D sequences taken through the z-stack.

The proposed technique could potentially be used for a wide range of motion tracking applications ranging from biological cells to articulated objects where the objects or the connection between the objects can be represented as a network. For instance, the technique can be used for tracking of bone marrow cells, peripheral blood cells and articulated vehicles such as trucks.

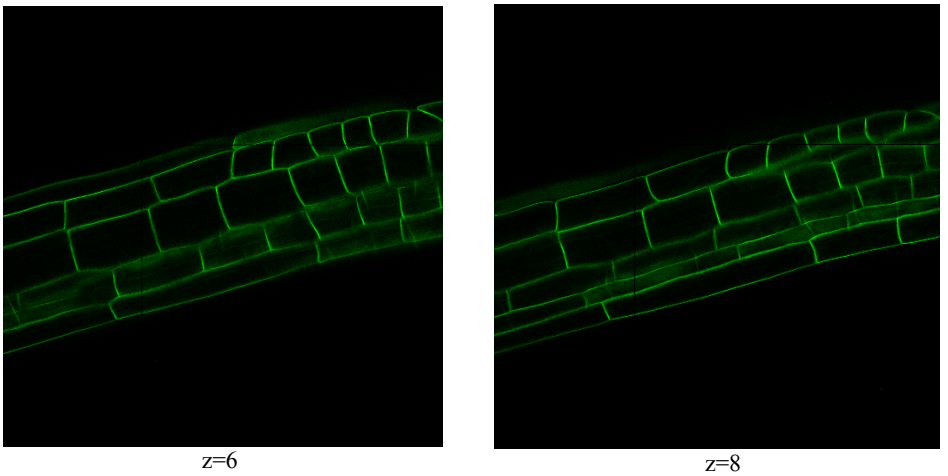
8.2. Future work

An advantage of the network-based approach is that the form of the output data structure mirrors the geometry of the underlying real life data. The output of the methodology is a network structure, which implicitly stores information about the geometry and arrangement of cells, as well as their movement. The resulting network snakes structure could be used to guide further processing. For example, knowing the location and boundary information about a cell, we could concentrate the search for particular features, such as marked nuclei or other sub cellular structures, to these areas. Alternatively, knowledge of the expected structure of the imaged root could be used to classify network components, recognising specific cell types.

Although several contributions were made in this thesis and the goals of this thesis have been achieved, some issues still remain unsolved and are recommendations for the future work as follows.

This thesis has concentrated on tracking cells in the elongation zone of the Arabidopsis root. New cells are formed by cell division surrounding the quiescent centre in the meristem as the plant grows. The change in dimensionality and topology of the network of cells caused due to cell division is a clear topic for future research. Having incorporated the Reversible Jump MCMC algorithm which can effectively deal with varying state dimensions, a detector that can detect dividing cells is what is required to drive the RJMCMC tracker with suitable proposals. Knowing the location and boundary information about cells that are being tracked by the NS-RJMCMC MRF tracker, each of these cells around the quiescent centre can then be examined at each frame for dividing cells that would have caused a sharp change in the characteristics such as the average image intensity of a cell. While useful cues to cell division may exist, it seems likely that the RJMCMC step will have a significant role to play in managing false positive node detections.

The capability of the proposed hybrid approach on 3D sequences taken through the z-stack has been shown in the evaluation section of the thesis. But there is a limitation in the approach to be dealt with. Some sequences taken through the z-stack show a complete change in the topological arrangement of cells which occurs when the growing root twists. One such example is shown in Figure 49, where the topological arrangement of cells at time *t=8* is completely different from time step *t=6*.



z=6 z=8

Figure 49. Drastic topological changes in a 3D sequence taken through a z-stack of CLSM images

This change in the network topology is too drastic and rapid for the tracker to automatically change its topology. This could also occur rarely in time based sequences if the twisting of the root is large enough to cause a shift in the chosen 2D plane to form the time based sequence. In such situations, the network topology needs reinitialising. Occurrences of this situation during tracking can be identified when the performance of the tracker suddenly gets worse indicating a need in the reinitialisation of the network topology. However, if the topology is reinitialised, the tracker is set to track a new network of cells. In other words, the tracker loses its previous state which is undesirable. Future work should be directed towards this complicated problem.

References

[1] V. Smil, “Phosporous in the Environment: Natural Flows and Human Interferences,” *Annual Review of Energy and the Environment*, vol. 25, Nov. 2000, pp. 53-88.

[2] D. Cordell, J.-O. Drangert, and S. White, “The story of phosphorus: Global food security and food for thought,” *Global Environmental Change*, vol. 19, May. 2009, pp. 292-305.

[3] H. Zhong and A. Läuchli, “Spatial and Temporal Aspects of Growth in the Primary Root of Cotton Seedlings: Effects of NaCl and CaCl2,” *Journal of Experimental Botany*, vol. 44, 1993, pp. 763 - 771.

[4] H. Ishikawa and M. Evans, “Induction of Curvature in Maize Roots by Calcium or by Thigmostimulation,” *Plant physiology*, vol. 100, 1992, pp. 762-768.

[5] A. French, S. Ubeda-Tomás, T.J. Holman, M.J. Bennett, and T. Pridmore, “High-throughput quantification of root growth using a novel image-analysis tool.,” *Plant physiology*, vol. 150, Aug. 2009, pp. 1784-95.

[6] J. Arsenault, S. Pouleur, C. Messier, and R. Guay, “WinRHIZO™, a root-measuring system with a unique overlap correction method,” *HortScience*, vol. 30, 1995, p. 906.

[7] P. Armengaud, K. Zambaux, A. Hills, R. Sulpice, R.J. Pattison, M.R. Blatt, and A. Amtmann, “EZ-Rhizo: integrated software for the fast and accurate measurement of root system architecture,” *Plant Journal*, vol. 57, 2008, pp. 945 - 956.

[8] J.L. Bot, V. Serra, J. Fabre, X. Draye, S. Adamowicz, and L. Pagès, “DART: a software to analyse root system architecture and development from captured images,” *Plant and Soil*, vol. 326, 2010, pp. 261 - 273.

[9] N.D. Miller, B.M. Parks, and E.P. Spalding, “Computer-vision analysis of seedling responses to light and gravity,” *The Plant Journal*, vol. 52, 2007, pp. 374 - 381.

[10] M.J. Jaffe, A.H. Wakefield, F. Telewski, E. Gulley, and R. Biro, “Computer-assisted image analysis of plant growth, thigmomorphogenesis and gravitropism.,” *Plant physiology*, vol. 77, Jan. 1985, pp. 722-730.

[11] B. Péret, A. Larrieu, and M.J. Bennett, “Lateral root emergence: a difficult birth,” *Journal of Experimental Botany*, vol. 60, 2009, pp. 3637 - 3643.

[12] B. Biskup, H. Scharr, A. Fischbach, A. Wiese-Klinkenberg, U. Schurr, and A. Walter, “Diel Growth Cycle of Isolated Leaf Discs Analyzed with a Novel, High-Throughput Three-Dimensional Imaging Method Is Identical to That of Intact Leaves,” *Plant Physiology*, vol. 10, 2009.

[13] M. Jansen, F. Gilmer, B. Biskup, K.A. Nagel, U. Rascher, A. Fischbach, S. Briem, G. Dreissen, S. Tittmann, S. Braun, I.D. Jaeger, M. Metzlaff, U. Schurr, H. Scharr, and A. Walter, “Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via GROWSCREEN FLUORO allows detection of stress tolerance in Arabidopsis thaliana and other rosette plants,” *Functional Plant Biology*, vol. 36, 2009, p. 902 – 914.

[14] B. Max, S. Vincent, S. Raju, R. Anne, T. Johan, G. Petter, J. Stefan, and S. Nathaniel, “LAMINA: a tool for rapid quantification of leaf size and shape parameters,” *BMC Plant Biology*, vol. 8, 2008, p. 82.

[15] R.V.L. Joosen, J. Kodde, L.A.J. Willems, W. Ligterink, and H.W.M.H. Linus H. W. Van Der Plas, “Germinator: a software package for high-throughput scoring and curve fitting of Arabidopsis seed germination,” *The Plant Journal*, vol. 62, 2010, pp. 148 - 159.

[16] M. Marcuzzo, P. Quelhas, A. Campilho, A.M. Mendonça, and A. Campilho, “A Hybrid Approach for Arabidopsis Root Cell Image Segmentation,” *Image Analysis and Recognition*, vol. 5112, 2008, pp. 739-749.

[17] L. Dupuy, J. Mackenzie, and J. Haseloff, “Coordination of plant cell division and expansion in a simple morphogenetic system,” *Proceedings of the National Academy of Sciences of USA*, vol. 107, 2010.

[18] D.P. McCullough, P.R. Gudla, B.S. Harris, J.A. Collins, K.J. Meaburn, M.A. Nakaya, T.P. Yamaguchi, T. Misteli, and S.J. Lockett, “Segmentation of Whole Cells and Cell Nuclei From 3-D Optical Microscope Images Using Dynamic Programming,” *IEEE Transactions on Medical Imaging*, vol. 27, 2008, pp. 723 - 734.

[19] T.I. Baskin, with C.M.V.D. Weele, H.S. Jiang, K.K. Palaniappan, and V.B. Ivanov, Kannappan Palaniappan, “A New Algorithm for Computational Image Analysis of Deformable Motion at High Spatial and Temporal Resolution Applied to Root Growth. Roughly Uniform Elongation in the Meristem and Also, after an Abrupt Acceleration, in the Elongation Zone,” *Plant Physiology*, vol. 132, 2003, pp. 1138 - 1148.

[20] T.J. Roberts, S.J. McKenna, C.-J. Du, N. Wuyts, T. a Valentine, and a G. Bengough, “Estimating the motion of plant root cells from in vivo confocal laser scanning microscopy images,” *Machine Vision and Applications*, vol. 21, Jul. 2009, pp. 921-939.

[21] T. Roberts, S. Mckenna, N. Wuyts, T. Valentine, and A. Bengough, “Performance of Low-Level Motion Estimation Methods for Confocal Microscopy of Plant Cells in vivo,” *IEEE Workshop on Motion and Video Computing*, 2007.

[22] M. Butenuth, “Segmentation of imagery using network snakes,” *Proc. Photogrammetric Computer Vision*, 2006.

[23] Z. Khan, T. Balch, and F. Dellaert, “An MCMC-based Particle Filter for Tracking Multiple Interacting Targets,” *Lecture Notes in Computer Science*, vol. 3024, 2004, pp. 279 - 290.

[24] Z. Khan, T. Balch, and F. Dellaert, “MCMC-based particle filtering for tracking a variable number of interacting targets,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, Nov. 2005, pp. 1805 - 1918.

[25] Ü. Kolukisaoglu and K. Thurow, “Future and frontiers of automated screening in plant sciences,” *Plant Science*, vol. 178, 2010, pp. 476 - 484.

[26] T.J. Roberts, S.J. McKenna, J. Hans, T. a Valentine, and a G. Bengough, “Part-Based Multi-Frame Registration for Estimation of the Growth Of Cellular Networks in Plant Roots,” *18th International Conference on Pattern Recognition (ICPR’06)*, 2006, pp. 378-381.

[27] N.R. PAL and S.K. PAL, “A Review on image segmentation techniques,” *Pattern Recognition*, vol. 26, 1993, pp. 1277 - 1294.

[28] J. Beutel, H.L. Kundel, and R.L.V. Metter, *Handbook of Medical Imaging*, Bellingham, WA, 2001.

[29] R.C. Gonzalez, Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[30] M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging*, vol. 13, 2004, pp. 146 - 165.

[31] P.L. Rosin, “Unimodal Thresholding,” *Pattern Recognition*, vol. 34, 2001, pp. 2083 - 2096.

[32] M.E. Dunn and S.H. Joseph, “Processing poor quality line drawings by local estimation of noise,” *Pattern Recognition, Lecture Notes in Computer Science*, vol. 301, 1988, pp. 153 - 162.

[33] J.S. Weszka, “A survey of threshold selection techniques,” *Computer Graphics and Image Processing*, vol. 7, 1978, pp. 259 - 265.

[34] C. Lee, S. Huh, T.A. Ketter, and M. Unser, “Unsupervised connectivity-based thresholding segmentation of midsagittal brain MR images,” *Computers in biology and medicine*, vol. 28, May. 1998, pp. 309-38.

[35] F. Rubin, “The Lee Path Connection Algorithm,” *IEEE Transactions on Computers*, vol. c-23, 1974.

[36] K. Wu, D. Gauthier, and M.D. Levine, “Live cell image segmentation.,” *IEEE transactions on bio-medical engineering*, vol. 42, Jan. 1995, pp. 1-12.

[37] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, 1975.

[38] S. Tabbone and D. Ziou, “Edge Detection Techniques - An Overview,” *International Journal on Pattern Recognition and Image Analysis*, vol. 8, 1998, pp. 537 - 559.

[39] L.G. Roberts, “Machine perception of three dimensional solids,” *Optical and Electro-optical Information Processing, MIT Press*.

[40] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, Nov. 1986, pp. 679-698.

[41] D. Marr and E. Hildreth, “Theory of edge detection.,” *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, vol. 207, Feb. 1980, pp. 187-217.

[42] L. Ding and A. Goshtasby, “On the Canny edge detector,” *Pattern Recognition*, vol. 34, Mar. 2001, pp. 721-725.

[43] H.S. Neoh and A. Hazanchuk, “Adaptive Edge Detection for Real-Time Video Processing using FPGAs,” *Global Signal Processing Conference*, Feb. 2004.

[44] W.Y. Ma and B.S. Manjunath, “Edge flow: A framework of boundary detection and image segmentation,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 744-749.

[45] B. Bhanu, S. Lee, C.C. Ho, and T. Henderson, “Range data Processing: Representation of surfaces by edges,” *Proc. 8th Int. Conf. on Pattern Recognition*, 1986, pp. 236 - 238.

[46] M. a Wani and B.G. Batchelor, “Edge-region-based segmentation of range images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, Mar. 1994, pp. 314-319.

[47] S. Beucher, “The watershed transformation applied to image segmentation,” *Scanning Microscopy - Supplement*, 1992, p. 299–314.

[48] N. Malpica, C.O. de Solórzano, J.J. Vaquero, A. Santos, I. Vallcorba, J.M. García-Sagredo, and F. del Pozo, “Applying watershed algorithms to the segmentation of clustered nuclei.,” *Journal of the International Society for Analytical Cytology*, vol. 28, Aug. 1997, pp. 289-97.

[49] G. Lin, U. Adiga, K. Olson, J.F. Guzowski, C. a Barnes, and B. Roysam, “A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks.,” *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, vol. 56, Nov. 2003, pp. 23-36.

[50] L. Najman and M. Schmitt, “Geodesic saliency of watershed contours and hierarchical segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, 1996, pp. 1163-1173.

[51] H.T. Nguyen, M. Worring, and R.V.D. Boomgaard, “Watersnakes : Energy-Driven Watershed Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, Mar. 2003, pp. 330-342.

[52] A. Albiol, L. Torres, and E.J. Delp, “An unsupervised color image segmentation algorithm for face detection applications,” *Proceedings International Conference on Image Processing*, 2001, pp. 681-684.

[53] M. Baccar, L.A. Gee, R.C. Gonzalez, and M.A. Abidi, “Segmentation of Range Images Via Data Fusion and Morphological Watersheds,” *Pattern Recognition*, vol. 29, Oct. 1996, pp. 1673-1687.

[54] M. Kass, A. Witkin, and D. Terzopolous, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 5, Jan. 1988, pp. 545-331.

[55] N. Ray and S.T. Acton, “Active Contours for Cell Tracking,” *IEEE Southwest Symposium on Image Analysis and Interpretation*, 2002, p. 274.

[56] H.S. Horace and D. Shen, “An affine-invariant active contour model (AI-Snake) for model-based segmentation,” *Image and Vision Computing*, vol. 16, 1998, pp. 135-146.

[57] D.J. Williams and M. Shah, “A Fast algorithm for active contours and curvature estimation,” *CVGIP: Image Understanding*, vol. 55, 1992, pp. 14-26.

[58] C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén, and J.-C. Olivo-Marin, “Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing.,” *IEEE transactions on medical imaging*, vol. 21, Oct. 2002, pp. 1212-21.

[59] F. Leymarie and M.D. Levine, “Tracking Deformable Objects in the Plane Using an Active Contour Model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, 1993, pp. 617 - 634.

[60] T. Mcinerney and D. Terzopoulos, “Topologically Adaptable Snakes,” *Proc.oftheFifth Int. Conf. onComputerVision*, 1995, pp. 840-845.

[61] S. Bischoff and L.P. Kobbelt, “Snakes with topology control,” *The Visual Computer*, 2003.

[62] F.A. Velasco and J.L. Marroquin, “Growing snakes: active contours for complex topologies,” *Pattern Recognition*, vol. 36, Feb. 2003, pp. 475-482.

[63] W.M. Neuenschwander, P. Fua, L. Iverson, G. Székely, and O. Kübler, “Ziplock Snakes,” *International Journal of Computer Vision*, vol. 25, 1997, pp. 191 - 201.

[64] C. Xu and J.L. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 7, Jan. 1998, pp. 359-69.

[65] L.D. Cohen, “On Active Contour Models and Balloons,” *CVGIP: Image understanding*, vol. 53, 1991, pp. 211-218.

[66] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, Jan. 1997, pp. 61 - 79.

[67] T.F. Chan and L.A. Vese, “Active contours without edges,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, Jan. 2001, pp. 266-77.

[68] F. Yang, M.A. Mackey, F. Ianzini, G. Gallardo, and M. Sonka, “Cell segmentation, tracking, and mitosis detection using temporal context,”

Medical image computing and computer-assisted intervention : MICCAI, vol. 3749/2005, Jan. 2005, pp. 302-309.

[69] K. Li and T. Kanade, "Cell Population Tracking and Lineage Construction Using Multiple-Model Dynamics Filters and Spatiotemporal Optimization," *Workshop on Microscopic Image Analysis*, 2007.

[70] S. Osher and J.A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, 1988, pp. 12-49.

[71] M. Liu, R.K. Yadav, A. Roy-Chowdhury, and G.V. Reddy, "Automated tracking of stem cell lineages of Arabidopsis shoot apex using local graph matching.," *The Plant journal : for cell and molecular biology*, vol. 62, Apr. 2010, pp. 135-47.

[72] C. Xu, "Generalized gradient vector flow external forces for active contours," *Signal Processing*, vol. 71, Dec. 1998, pp. 131-139.

[73] A. Amini, S. Tehrani, and T.E. Weymouth, "Using Dynamic Programming For Minimizing The Energy Of Active Contours In The Presence Of Hard Constraints," *[1988 Proceedings] Second International Conference on Computer Vision*, 1988, pp. 95-99.

[74] Y.S. Akgul and C. Kambhamettu, "A Scale-Space Based Approach for Deformable Contour Optimization," *Lecture Notes in Computer Science*, vol. 1682, 1999, pp. 410-421.

[75] J.A.M. Beliën, H.A.H.M.V. Ginkel, P. Tekola, L.S. Ploeger, N.M. Poulin, J.P.A. Baak, and P.J.V. Diest, "Confocal DNA cytometry: A contour-based segmentation algorithm for automated three-dimensional image segmentation," *Journal of the International Society for Advancement of Cytometry*, vol. 49, 2002, pp. 12 - 21.

[76] V. Sethuraman, S. Taylor, A. French, D. Wells, and T. Pridmore, "Segmentation and Tracking of Confocal Images of Arabidopsis thaliana Root Cells using Automatically- Initialized Network Snakes," *3rd International Conference on Bioinformatics and Biomedical Engineering*, 2009, pp. 1-4.

[77] N. Peterfreund, "The Velocity Snake," *Proc. IEEE Nonrigid and Articulated Motion Workshop*, 1997.

[78] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A Survey," *ACM Computing Surveys*, vol. 38, Dec. 2006, p. 13-es.

[79] H.A. Rowley, S. Baluja, and T. Kanade, "Human Face Detection in Visual Scenes," *Advances in Neural Information*, 1995.

[80] G.D. Croon and E.O. Postma, “Active object detection,” *2nd International Conference on Computer Vision Theory and Applications*, 2007, pp. 97-103.

[81] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, “Object detection and localization using local and global features,” *Towards Category-Level Object Recognition*, 2005.

[82] R. Pereira and L.S. Lopes, “Learning Visual Object Categories with Global Descriptors and Local Features,” *Progress in Artificial Intelligence*, vol. 5816, 2009, pp. 225-236.

[83] R.E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transaction of the ASME-Journal of basic Engineering*, vol. 82, 1960, pp. 35-45.

[84] M. Isard and A. Blake, “CONDENSATION—Conditional Density Propagation for Visual Tracking,” *International Journal of Computer Vision*, vol. 29, 1998, pp. 5-28.

[85] M. Isard and A. Blake, “A mixed-state condensation tracker with automatic model-switching,” *Sixth International Conference on Computer Vision*, 1998, pp. 107-112.

[86] H.A.P. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients,” *IEEE Transactions on Automatic Control*, vol. 33, 1988, pp. 780 - 783.

[87] W. Ng, J. Li, S. Godsill, and J. Vermaak, “Tracking Variable Number of Targets using Sequential Monte Carlo Methods,” *13th Workshop on Statistical Signal Processing,*, 2006.

[88] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, “Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization,” *IEEE Computer Society conference on Computer vision and pattern recognition*, vol. 2, 1999, p. 594.

[89] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” *Technical Report TR 95-041, University of North Carolina*, 1995.

[90] D. Beymer and K. Konolige, “Real-Time Tracking of Multiple People Using Continuous Detection,” *Proceedings of the IEEE Frame Rate Workshop*, 1999.

[91] N. Peterfreund, “Robust tracking of position and velocity with Kalman snakes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, Jun. 1999, pp. 564-569.

[92] N. Ray, S.T. Acton, and K. Ley, “Tracking leukocytes in vivo with shape and size constrained active contours.,” *IEEE transactions on medical imaging*, vol. 21, Oct. 2002, pp. 1222-35.

[93] D. Tweed and A. Calway, “Tracking Many Objects Using Subordinated Condensation,” *British Machine Vision conference*, 2002, pp. 283-292.

[94] H. Kang and D. Kim, “Real-time multiple people tracking using competitive condensation,” *Pattern Recognition*, vol. 38, Jul. 2005, pp. 1045-1058.

[95] D. Schulz, D. Fox, and A.B. Cremers, “Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association,” *IEEE Computer Society conference on Computer vision and pattern recognition*, vol. 1, 2001, pp. 371 - 377.

[96] D. Adele and G. Whitford, “Particle Filter Based Algorithm for Target Position Estimation Under Sparce Sensor Surveillance,” *Digital Signal Processing Workshop*, 2006, pp. 482-487.

[97] M.J. Black and A.D. Jepson, “A Probabilistic Framework for Matching Temporal Trajectories: Condensation-Based Recognition of Gestures and Expressions,” *Lecture notes in Computer Science*, vol. 1406, 1998, p. 909.

[98] I. Smal, W. Niessen, and E. Meijering, “Particle Filtering for Multiple Object Tracking in Molecular Cell Biology,” *IEEE Nonlinear Statistical Signal Processing Workshop*, 2006, pp. 129 - 132.

[99] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, 2002, pp. 174 - 188.

[100] Y. Zhai and M. Yeary, “Implementing particle filters with Metropolis-Hastings algorithms,” *Region 5 Conference: Annual Technical and Leadership Workshop, 2004*, pp. 149-152.

[101] K. Smith, S.O. Ba, J.-M. Odobez, and D. Gatica-Perez, “Tracking the visual focus of attention for a varying number of wandering people,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, Jul. 2008, pp. 1212-1229.

[102] L.S. Shapiro, H. Wang, and J.M. Brady, “A Matching and Tracking Strategy for Independently Moving Objects,” *Proceedings of the British Machine Vision Conference*, 1992, p. 306–315.

[103] F.P. Ferrie, M.D. Levine, and S.W. Zucker, “Cell Tracking: A Modeling and Minimization Approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, May. 1982, pp. 277-291.

[104] J. Bulas-Cruz, A.T. Ali, and E.L. Dagless, “A Temporal Smoothing Technique for Real-Time Motion Detection,” *Computer Analysis of Images and Patterns*, vol. 719, 1993, pp. 379-386.

[105] J. Nash, J. Carter, and M. Nixon, “Dynamic feature extraction via the velocity Hough transform,” *Pattern Recognition Letters*, vol. 18, Oct. 1997, pp. 1035-1047.

[106] T.J. Holman, M.H. Wilson, K. Kenobi, I.L. Dryden, T.C. Hodgman, A.T. Wood, and M.J. Holdsworth, “Statistical evaluation of transcriptomic data generated using the Affymetrix one-cycle, two-cycle and IVT-Express RNA labelling protocols with the Arabidopsis ATH1 microarray.,” *Plant methods*, vol. 6, Jan. 2010, p. 9.

[107] L. Vincent and P. Soille, “Watersheds in Digital Spaces : An Efficient Algorithm Based on Immersion Simulations,” *IEEE Trans.on Pattern Matching and Machine Intelligence*, vol. 13, 1991, pp. 583 - 598.

[108] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, “Ridges for image analysis,” *Journal of Mathematical Imaging and Vision*, vol. 4, Dec. 1994, pp. 353-373.

[109] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density,” *In Proc. European Conf. Computer Vision*, vol. 1064, 1996, pp. 343 - 356.

[110] W.K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, 1970, pp. 97-109.

[111] L. Tierney, “Markov Chains for Exploring Posterior Distributions,” *The Annals of Statistics*, vol. 22, 1994, pp. 1701-1728.

[112] K. Smith, S.O. Ba, D. Gatica-Perez, and J.-M. Odobez, “Tracking the multi person wandering visual focus of attention,” *Proceedings of the 8th international conference on Multimodal interfaces - ICMI ’06*, 2006.

[113] F. Bashir and F. Porikli, “Performance Evaluation of Object Detection and Tracking Systems,” *IEEE International Workshop on Performance Evaluation of Tracking andSurveillance (PETS)*, 2006.

[114] F. Yin, D. Makris, and S. Velastin, “Performance Evaluation of Object Tracking Algorithms,” *Proceeding Tenth IEEE InternationalWorkshop on Performance Evaluation of Tracking and Surveil-lance*, 2007.

[115] R. Spangenberg and D. Thomas, “Evaluation of object tracking in traffic scenes,” *ISPRS,Commission V Symposium, ImageEngineering and Vision Metrology*, 2006.

[116] K. Smith, D. Gatica-perez, J.-marc Odobez, and S. Ba, “Evaluating Multi-Object Tracking,” *Proc. IEEE CVPR Workshop on Empirical Evaluation Methods on Computer Vision*, 2005.

[117] L.M. Brown, A.W. Senior, Y.-li Tian, J. Connell, A. Hampapur, C.-fe Shu, H. Merkl, and M. Lu, “Performance Evaluation of Surveillance Systems Under Varying Conditions,” *IEEE International Workshop on Performance Evaluation of Tracking andSurveillance (PETS)*, 2005.

[118] A.F. Clark and C. Clark, “Performance Characterization in Computer Vision A Tutorial,” *In http://www.peipa.essex.ac.uk/benchmark/, Essex, UK*, Mar. 1999.

[119] J. Black, T. Ellis, and P. Rosin, “A Novel Method for Video Tracking Performance Evaluation,” *IEEE International Workshop on Performance Evaluation of Tracking andSurveillance (PETS)*, 2003.

[120] C. Jaynes, S. Webb, R.M. Steele, and Q. Xiong, “An Open Development Environment for Evaluation of Video Surveillance Systems,” *Proceedings of the Third International Workshop on Performance Evaluation of Tracking and Surveillance (PETS’2002),*, 2002.

[121] H. Sidenbladh and S.-L. Wirkander, “Tracking Random Sets of Vehicles in Terrain,” *Conference on Computer Vision and Pattern Recognition Workshop*, Jun. 2003, pp. 98-98.

[122] H. Sidenbladh, “Multi-Target Particle Filtering for the Probability Hypothesis Density,” *International Conference on Information Fusion*, 2003.

[123] B. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo Implementation of the PHD Filter for Multi-target Tracking,” *Proceedings of the Sixth International Conference of Information Fusion*, 2003.

[124] P.J. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, 1995, pp. 711-732.